A NEW CONSTITUTIVE MODEL FOR THE FINITE ELEMENT ANALYSIS OF METAL POWDER COMPACTION

BY
P.G. MARAIS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering

Department of Mechanical Engineering
University of Cape Town
Rondebosch 7700
South Africa

September 1996

DECLARATION

This is to certify that the work presented in this thesis is essentially my own, and that no part of it has been submitted for a degree at this or any other university.

P. G. Marais
September 1996

ACKNOWLEDGEMENTS

I wish to express my gratitude to the following people:

- My supervisor, Dr Greg Mitchell.
- My colleagues and friends at CERECAM.
- My family and friends for their continuous support and encouragement.

I would also like to thank the FRD and CERECAM for their financial support.

SYNOPSIS

During the commercial compaction of metal powder in a punch-die setup and the subsequent sintering of the compacted preform, the large confining pressures, friction and heat that arise determine the properties of the final component. An important application of the modelling of metal powder compaction is the determination of the properties of the compacted preform as these affect the properties of the final sintered component. An uneven density distribution in the compacted preform, for example, can cause cracking in the component during the sintering stage. The mechanical behaviour of the metal powder during compaction largely determines the properties of the compacted preform.

To date the consolidation of metal powders has been conveniently represented using constitutive theories based on elastic-plastic material models. Most of the work has concentrated on the use of *quadratic* yield surfaces but experimentation has shown that this type of yield surface does not always correctly represent metal powder behaviour (Brown, 1994). This thesis details the development of a new constitutive model for the finite element analysis of metal powder compaction based on extensive experimental testing done by Watson(1993) on aluminium powder. Watson found that a *cap* yield surface, often used in soil plasticity, best fitted the experimental data obtained from the aluminium powder. A model similar to that proposed by Watson was implemented in this thesis in an attempt to show that a *cap* yield surface is more accurate for modelling the compaction of aluminium powder and other powders than a *quadratic* yield surface.

The new constitutive model combines a Drucker-Prager or shear yield surface and a density evolving *cap* or consolidation yield surface to model powder compaction and differs from other *cap* models as the shear yield surface also evolves with density. A combination of an associative flow rule on the consolidation yield surface and a von Mises flow rule on the shear yield surface made for easier numerical implementation of the model. The model was implemented in ABAQUS as a FORTRAN 77 User-Material Subroutine using an Euler Backward integration scheme.

The model was calibrated and tested for -100 mesh aluminium powder and Hoeganaes ASC100.29 iron powder. Two simulations of the compaction of an iron ring were also run. The second simulation was run for comparison purposes using the Porous Metal Plasticity Model that is available in ABAQUS which is similar to a *quadratic* yield surface model.

It was found that the new constitutive model agreed well with experimental results for hydrostatic and constrained compression of aluminium and iron powder, especially for relative densities below 0.9. Deviation from experiment above a relative density of 0.9 was attributed to simplifications made to the consolidation yield surface at the constrained compression point or the model's inability to model compaction at relative densities approaching full density. Tests would have to be done to determine the actual reason. Similar results were also observed for the ring compaction simulation using the

new constitutive model and it was found to be more suitable than the *quadratic* model for modelling the compaction of iron powder.

The new constitutive model developed in this thesis can therefore be used with some confidence for the finite element analysis of the compaction of -100 mesh aluminium powder and ASC100.29 iron powder but more testing should be done using other metal powders and more complicated die geometries.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

FIGURES

TABLES

NOMENCLATURE

SYMBOLS

Special Symbols

$\dot{\bullet}$	differential with respect to time
$\underline{\ }$	vector
$\underline{\underline{\ }}$	matrix
T (superscript)	transpose of a vector
d	differential with respect to
∂	partial differential with respect to

Lower Case Characters

a-c	material parameters for new constitutive model
b_K, b_G	terms in general elastic moduli equations
$\underline{e}$	deviatoric strain vector
e_{ij}	deviatoric strain tensor
f_α	yield surface
g_α	plastic potential
k	material constant
$k(\rho)$	density dependent variable for new constitutive model
q_1-q_3	material parameters for Porous Metal Plasticity Model

Upper Case Characters

C	cohesion
C_i	Watson's yield surface variables
$C_i(\rho)$	density dependent variables for new constitutive model
$\underline{\underline{C}}$	deviatoric stress to stress transformation matrix
$\underline{\underline{C}}_s$	strain to deviatoric strain transformation matrix
$\underline{\underline{C}}_{sa}$	stress to deviatoric stress transformation matrix
$\underline{\underline{D}}_{el}$	elastic matrix
$\underline{\underline{D}}_{tan}$	consistent modulus
E	Young's modulus
G	fully dense shear modulus
$G(\rho)$	density dependent shear modulus
$\underline{\underline{G}}$	deviatoric elastic matrix
I_1	first stress invariant
S	second invariant of the deviatoric stress tensor

$\underline{S}$	deviatoric stress vector
S_{ij}	deviatoric stress tensor

Greek Characters

α	material constant
$\alpha_N(\rho)$	density dependent variable for new constitutive model
Δ	increment in
$\underline{\varepsilon}$	strain vector
ε_{ij}	strain tensor
ε_v	volumetric strain
φ	internal angle of friction
κ	hardening parameter
λ_α	plastic multiplier
λ	Lame's constant
μ	Lame's constant
ν	Poisson's ratio
ρ	relative density
ρ_0	initial relative density
$\underline{\sigma}$	stress vector
σ_{ij}	stress tensor
σ_m	hydrostatic stress
σ_y	yield stress
τ	shear stress

Subscripts

i,j,k,l	tensorial indices
n	increment number
α	active yield surface(shear,cons)

Superscripts

d	material parameters for new constitutive model
e, p	elastic, plastic components
E	elastic predictor
f-h	material parameters for new constitutive model
i	i-th iteration
j	material parameter for new constitutive model
k	material parameter for new constitutive model
k	k-th iteration
λ_1, λ_2	Watson's elastic moduli parameters
μ_1, μ_2	Watson's elastic moduli parameters

CHAPTER 1

INTRODUCTION

The manufacturing of engineering components by means of the metal powder compaction process consists of the compressing of a metal powder in a die under high confining pressures to produce what is referred to as a preform. This compacted preform is then sintered to obtain the final product. A wide variety of products are manufactured by this method which has the advantage over other manufacturing processes that the products often require little additional finishing.

Metal powder compaction can be divided into two main types: uniaxial or constrained compression compaction and cold isostatic compaction. Constrained compression of a metal powder involves the compaction of the powder in a punch and die setup. During this process powder is fed into a die and is then compacted by one or more punches to produce the preform. This method of compaction is used to produce medium-sized compacts and can be setup so that the sintered component can be produced with little additional shaping needed to obtain the finished product. An example of a punch and die setup is shown schematically in Figure 6.1.

An alternative form of compaction which is used for the compaction of larger compacts is cold isostatic compaction. During this process the component is subjected to equal pressure from all directions, instead of along just one axis, as for constrained compression. This allows larger powder sections to be pressed with a uniform density which produces more uniform shrinkage during sintering (Brookes, 1992:19). Metal powder is compacted isostatically by placing it in a rubber bag. The bag is then placed in a pressure chamber and hydraulic pressure is applied. Additional shaping of the final sintered component produced by isostatic pressing is often needed as the accuracy during compaction is not as high as constrained compression. Another form of isostatic compaction that is used is hot isostatic pressing. This method combines compaction and sintering in a single stage to produce components of high quality.

Aspects of the powder compaction process which must be addressed in modelling include friction between the powder and die, the large deformation that the powder undergoes, large stresses induced during compaction and the mechanical behaviour of the powder particles. Probably the biggest problem when modelling powder compaction is modelling the actual mechanical behaviour of the powder which behaves like a granular material at the beginning of compaction but more like a fully dense metal at the end of compaction.

An important aspect of the modelling of metal powder compaction is the determination of the final properties of the compacted preform as the properties of the preform affect the properties of the final sintered component. An uneven density distribution in the

compacted preform, for example, can cause cracking in the component during the sintering stage.

To date consolidation of metal powders has been conveniently represented using constitutive theories based on elastic-plastic material models. Most of the work has concentrated on the use of *quadratic* yield surfaces but experimentation has shown that this type of yield surface does not always correctly represent metal powder behaviour. This thesis details the development of a constitutive model for the finite element analysis of metal powder compaction based on extensive experimental testing done by Watson(1993) on aluminium powder. Watson found that a *cap* yield surface, often used in soil plasticity, best fitted the experimental data obtained from the aluminium powder. A model similar to that proposed by Watson was implemented in this thesis in an attempt to show that a *cap* yield surface is more accurate for modelling the compaction of aluminium powder and other powders than a *quadratic* yield surface.

The objectives of this thesis are therefore:

1. To develop the constitutive equations for the new constitutive model and implement them in a general purpose finite element code.
2. To verify the model by calibrating it for aluminium powder and comparing it with experimental results.
3. To calibrate the model for iron powder and compare the results of a finite element simulation of the compaction of an iron ring using the new model and an existing *quadratic* plasticity model.
4. To draw conclusions based on the results.

The implementation and testing of the new model was carried out using ABAQUS/Standard (HKS, 1995), a general purpose finite element program. The new constitutive model was verified using data obtained from available literature. Other data was obtained from the actual compaction of an iron ring (Kerr, 1994).

The remainder of this thesis is organised as follows: A review of the literature of existing constitutive models for modelling metal powder compaction and the testing of metal powder under compaction is given. The development of the new constitutive model is then given and is followed by details of the implementation of the new model in ABAQUS. The model is calibrated for aluminium powder and is verified using a number of tests. The model is then calibrated for iron powder and the results of a finite element simulation of the compaction of an iron ring using the new model and a Porous Metal Plasticity Model, available in ABAQUS, are compared. Finally, conclusions are drawn based on these results and recommendations are made.

CHAPTER 2

LITERATURE REVIEW AND DISCUSSION OF RELEVANT THEORY

In this chapter the theory of metal powder behaviour, the testing of this behaviour and finally the modelling of the constitutive behaviour of the powder is discussed by reviewing the available literature.

2.1 THE MECHANICAL BEHAVIOUR OF METAL POWDER UNDER COMPACTION

Metal powder granules differ in size and shape depending on the type of metal and the manufacturing process. Figure 2.1 shows an iron powder that was produced by hydrogen reduction of iron oxide. Iron powder particles have an irregular morphology that causes interlocking of the particles during compaction (Brown, 1994:388). Atomized aluminium particles also have an irregular morphology (Kamalie, 8).

Figure 2.2 shows a gas atomised copper powder that has almost spherical particles. The spherical shape and smaller size of these particles give the copper powder a higher loose relative density of 0.63 at the start of compaction compared to that of 0.38 for iron powder (Brown, 1994:391).

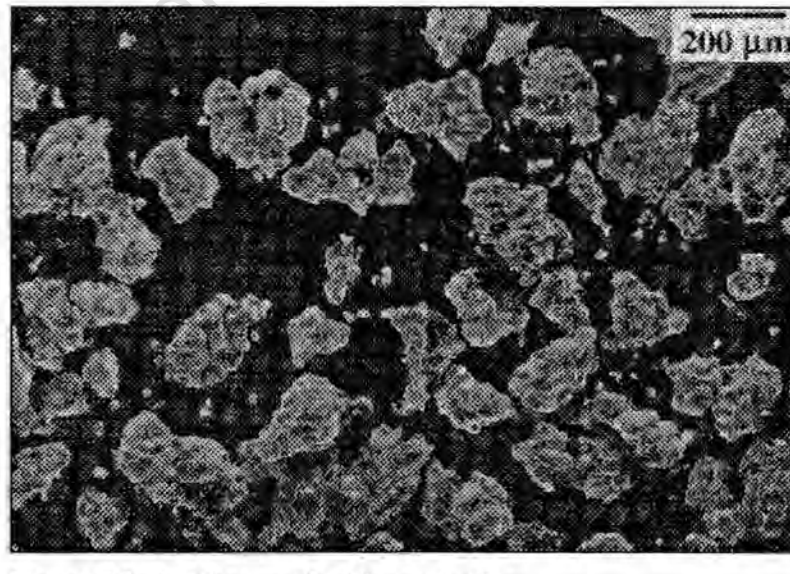

Figure 2.1: A micrograph of iron powder.

From experiments on iron and copper powders, Brown(1994:394) was able to establish that both the powder morphology and the initial conditions of the powder can affect

the behaviour of the powder. A particle which is more irregular will produce more interparticle cohesion which will, inturn, affect the consolidation characteristics of the powder.

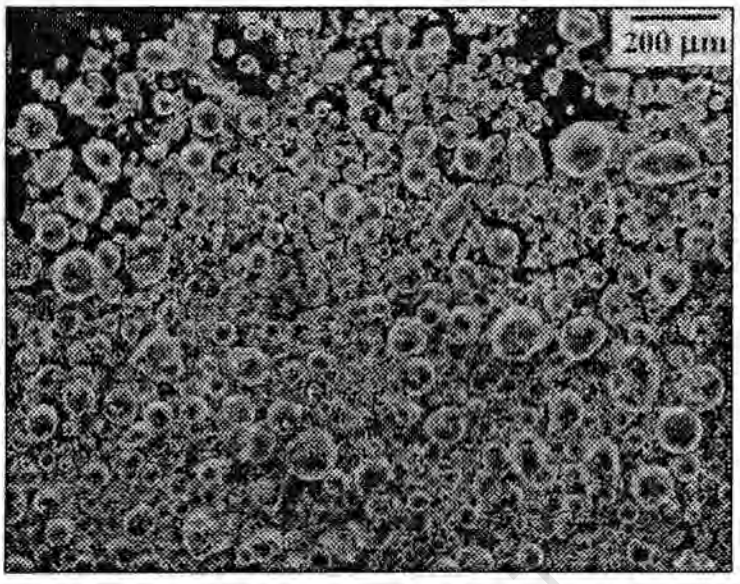

Figure 2.2: A micrograph of gas-atomised copper powder.

A metal powder which has little or no elastic stiffness, initially, therefore behaves like a granular material at the start of compaction. After some rearrangement of the particles has occurred they start to interact first elastically and then plastically and the morphology of the particles affect how they behave. Kamalie(9) stated that the particles will slide over each other and rearrange at low pressures for both hydrostatic and uniaxial loading.

The plastic deformation proceeds rapidly while the particle faces make contact and the voids between the particles collapse. Friction forces between the particles and between the particles and the die walls start to develop and increase as the compaction process continues. Lenel(1,7) stated that friction in the compact during compaction can cause uneven density distributions. An uneven density distribution can result in an uneven strength of the compacted preform and the final result of this could be cracking in the sintered product due to differential expansion and distortion (Haggblad, 1993:S2). Lubricant is usually added to the metal powder to reduce the effect of friction.

Another important property of a metal powder under compaction is its ability to flow (Lenel, 3) which is different to that of a fluid. A confined fluid will transmit a uniform stress even if it flows around a corner. A metal powder, however, can sustain a shear force and will therefore only flow in the direction of the applied pressure. This indicates the importance of shear stress in the consolidation of a metal powder. Watson(1993:2080) did extensive tests on aluminium powder and the results obtained indicated that the amount of hydrostatic stress needed for the consolidation of aluminium powder was reduced as more shear stress was introduced. Brown(1994:395) also observed that the yield behaviour of metal powder was affected by the deformation path. It was found that, although a powder sample consolidated by pure hydrostatic loading may have a higher relative density compared to a sample

consolidated by constrained compression, the constrained compression sample can have a higher yield stress. This was attributed to interparticle cohesion.

Therefore, during the plastic deformation stage of powder compaction, the particles flow plastically as a result of the applied pressure and cohesion. This will result in the collapsing of interparticle voids and this consolidation will increase the elastic stiffness of the powder body. As the relative density of the powder approaches one, so the elastic stiffness of the powder approaches that of the fully dense material.

It can therefore be seen that during the consolidation process, a metal powder makes the transition from a granular material at the start of compaction to a semi-consolidated or fully consolidated metal and this mechanical behaviour must be accurately modelled in a constitutive model if the final properties of a compacted preform are to be predicted accurately. Extensive testing using different deformation paths should be carried out so that the mechanical behaviour of the powder to be modelled is fully understood.

2.2 TESTING THE BEHAVIOUR OF METAL POWDER

Although many constitutive models have been developed for metal powder compaction, most of them are based on limited experimental data (Brown, 1994:383). The data obtained in many cases is from sintered porous bodies and not loose powders and often only one deformation path is considered. A few extensive studies of powder behaviour have been done and are discussed here.

The elastic moduli of the powder change during compaction. It is essential that the variation of the bulk and shear moduli with relative density is known for any constitutive model. Various methods for determining the bulk and shear moduli have been proposed. Ramakrishnan(1990:3930) proposed an equation for the determination of the effective elastic moduli using the relative density of the powder and the modulus concerned and Poisson's ratio of the fully dense material:

$$M^* = M\frac{\rho^2}{(1 + b_m(1-\rho))} \qquad (2.1)$$

where M^* is the effective modulus, M is the corresponding fully dense modulus and ρ is the relative density. The term b_m depends on the modulus concerned and is a function of the fully dense material's Poisson's ratio. This equation was validated using experimental data for porous MgO, $MgAl_2O_3$ and Sm_2O_3.

Watson(1993:2072) determined the bulk and shear moduli of aluminium powder during extensive tests on -100 mesh aluminium powder to develop a constitutive model for the powder. Ultrasonic methods were used to measure the shear and longitudinal wave velocities for samples at different relative densities. The dependence of Lame's

constants, λ and μ, on relative density was determined from this data and hence the elastic moduli can be calculated.

The other tests that Watson performed were also completed with the help of ultrasonic methods. Four simple tests were done on the aluminium powder to determine its yield behaviour. Constrained compression tests were performed on 15g samples using a single-acting 25.4mm diameter press. The samples were consolidated to varying relative densities using different loads. The ultrasonic methods were used to determine the density of the samples. The principal stresses, σ_1, σ_2, and σ_3 in the sample can be calculated as follows:

The axial stress in the sample, σ_3, is obtained from

$$\sigma_3 = (\lambda + 2\mu)\varepsilon_3^e \qquad (2.2)$$

where ε^e_3 is the axial elastic strain. The transverse stresses, σ_1 and σ_2, can be calculated from the elastic constitutive relation as

$$\sigma_1 = \sigma_2 = 2\mu\varepsilon_3^e = \frac{2\mu}{(\lambda + 2\mu)}\sigma_3 \qquad (2.3)$$

From this data, a point can be located in stress space at yield for a particular density and therefore information about the yield locus is known.

Watson(1993:2073) also performed hydrostatic compression tests using a hot isostatic press. The tests were performed at temperatures below 30°C and samples were prepared by placing 15g of aluminium powder in aluminium foil tubes. The tubes were evacuated and then crimped before pressing. These tests provided data of hydrostatic stress versus relative density.

The hydrostatic tests were followed by unconstrained tension and compression tests. The samples for these tests were prepared by electro-discharge machining of cylinders from the samples previously consolidated by either constrained compression or hydrostatic compression. The samples were 5mm in diameter and 10mm in height. Both tests were performed by placing the samples between steel platens on a servohydraulic test machine. The yield stress for both tests was determined when the load-displacement curves deviated from a linear relationship. The results of the four tests were then used to construct yield loci for different relative densities.

Schwartz(1969:81) conducted a similar study into the yield behaviour of -100 mesh hydrogen reduced iron powder under compressive loading. Hydrostatic and triaxial compression tests were performed using a pressure vessel and piston apparatus. The iron powder samples used in the tests were placed in copper jackets, 13mm in diameter. The tests were performed by first applying hydrostatic pressure and then applying hydrostatic loading with axial loading on the same samples. The hydrostatic

compression tests provided hydrostatic pressure versus density data and the triaxial tests provided data which could be used to determine the Mohr-Coulomb parameters dependence on relative density. The Mohr-Coulomb yield criterion is defined as

$$\tau = C + \sigma \tan\varphi \qquad (2.4)$$

where τ is the shear stress, σ is the normal stress and C and φ are the Mohr parameters. C is the cohesion of the powder and φ is the angle of internal friction. These parameters can be related to the Drucker-Prager yield criterion parameters. Again these tests provide yield loci behaviour for the iron powder.

Brown(1994:386) performed tests on both copper and iron powders in order to compare actual metal powder behaviour with that predicted by various constitutive models proposed for metal powder compaction. Gas atomised copper powder and hydrogen reduced Hoeganaes MH-100 iron powder were used in biaxial and triaxial compaction test systems. Both systems were displacement controlled devices and were designed to load the powders along proportional deformation paths. The biaxial system consisted of a cylindrical die with double-acting upper and lower punches and therefore performed constrained compression tests. The triaxial system compacted a cubed volume of powder using proportional displacements and was used for hydrostatic compression tests and other proportional loading tests. The relative displacement ratios used in these tests were 1:1:1, 1:0.4:0.4 and 1:0:0. The results of these tests were then plotted in deviatoric-hydrostatic stress space so that the powders' behaviour could be compared to that predicted by the constitutive models.

Brown(1994:388) indicated how friction could affect the results of these powder tests and appropriate precautions were taken to reduce the friction. A comparison between the results on iron powder done by Schwartz(1969:84) and Brown(1994:393) indicated that the results of hydrostatic stress produced by Schwartz were up to twice as much as those produced by Brown for the same relative density. No mention was made of the use of lubricants in Schwartz's work and this could be the reason for the difference. Changes in powder compactability since Schwartz's work could also be a reason for the difference. Some work has also been done on determining coefficients of friction for metal powders (Tabata, 1981:179) but will not be discussed here.

The mechanical behaviour of a metal powder can therefore be determined using a few simple laboratory tests. A combination of constrained compression tests and triaxial tests which allow hydrostatic loading are enough to determine the yield behaviour of a powder and allow the development of a constitutive model to model the powder behaviour.

2.3 CONSTITUTIVE MODELLING OF METAL POWDER BEHAVIOUR

Most of the constitutive models that have been proposed for modelling metal powder compaction are based on a yield surface with an elliptical shape and can be termed

quadratic (Watson, 1993:2071). Models developed from soil plasticity have also been used. Brown(1994:385) did indicate, however, that when different constitutive models were compared, care should be taken when deciding how to compare the models as density variation on its own did not appear to be a good arbiter. Most simulations matched relative density satisfactorily which may indicate that density variation is more dependent on the die geometry than on the constitutive behaviour of the powder. Constitutive models of the *quadratic* and soil plasticity or *cap* type which have been proposed for modelling powder compaction will be introduced in this section and compared with the behaviour that was obtained from experiments.

2.3.1 *Quadratic* Yield Surfaces

Doraivelu(1984:527) compared a number of different models with *quadratic* yield surfaces. The yield surfaces were all functions of the hydrostatic stress, σ_m, and the second invariant of the deviatoric stress tensor, S, where (using standard indicial notation)

$$\sigma_m = \tfrac{1}{3}\sigma_{kk} \qquad (2.5)$$

$$S = \sqrt{\tfrac{1}{2}S_{ij}S_{ij}}\,, \quad S_{ij} = \sigma_{ij} - \delta_{ij}\sigma_m \qquad (2.6)$$

where σ_{ij} is the Cauchy stress tensor. The *quadratic* yield surface can be represented in the form

$$A(\rho)S^2 + B(\rho)\sigma_m{}^2 - \delta(\rho)\sigma_y{}^2 = 0 \qquad (2.7)$$

where ρ is the relative density, $A(\rho)$, $B(\rho)$ and $\delta(\rho)$ are functions of relative density and σ_y is the compressive yield stress of the fully dense material.

This yield surface has an elliptical shape in stress space. Figure 2.3 illustrates the shape of the yield surface in deviatoric-hydrostatic stress space for various relative densities. It is characteristic for these yield surfaces to approach the von Mises yield surface as the relative density approaches one. Relative density is therefore used as a state variable to control the deformation history of the powder. Other characteristics are symmetry with respect to hydrostatic tension and compression and rotational symmetry about the hydrostatic axis (Watson, 1993:2071).

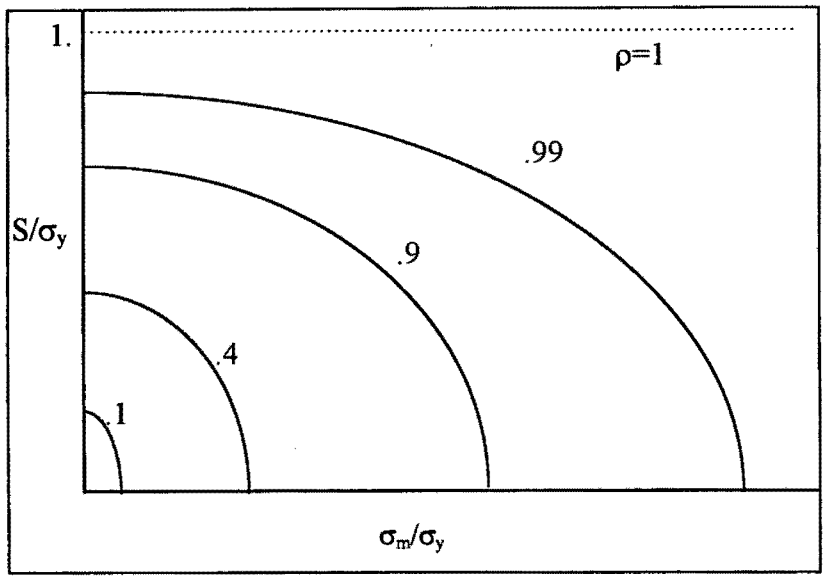

Figure 2.3: Quadratic yield surface at different relative densities.

It can also be seen from Figure 2.3 that the deviatoric stress decreases with increasing hydrostatic stress for a *quadratic* yield surface. From the results of tests on iron and copper powders, Brown(1994:394) found that this might not be appropriate for powder compaction, especially at low relative densities when the powder behaves more like a granular material. Brown's results are shown in Figure 2.4 and Figure 2.5. Dotted lines in both figures connect points of constant relative density and the symbols represent stress points on different deformation paths. It can be seen that initially the deviatoric stress increases with increasing hydrostatic stress. This was more evident for the copper powder than the iron powder but Brown stated that tests done by Gollion(1989) on tungsten powder and tests done by Watson(1993:2076) on aluminium powder also indicated the same trend. The differences displayed by iron powder were attributed to its irregular morphology which produced more interparticle cohesion even at low hydrostatic pressures. These differences could indicate that a single type of yield surface is inappropriate for modelling the compaction of all metal powders. Figure 2.4 clearly indicates that a quadratic yield surface, indicated by the solid line, is not appropriate for copper powder.

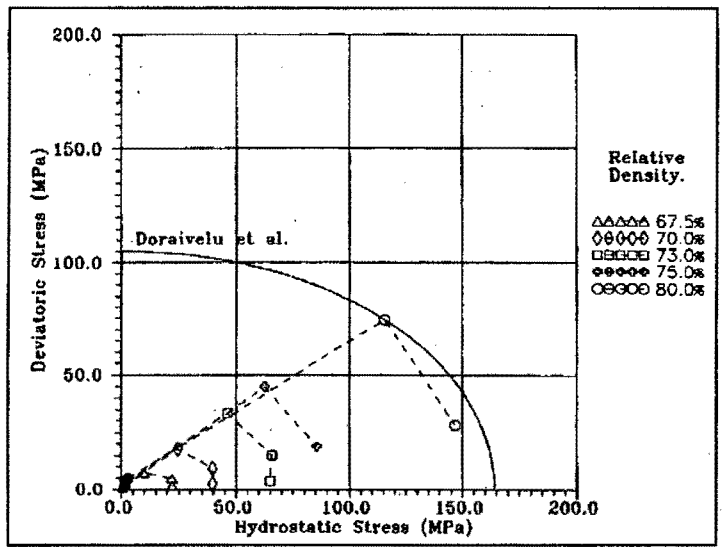

Figure 2.4: Brown's results for copper powder.

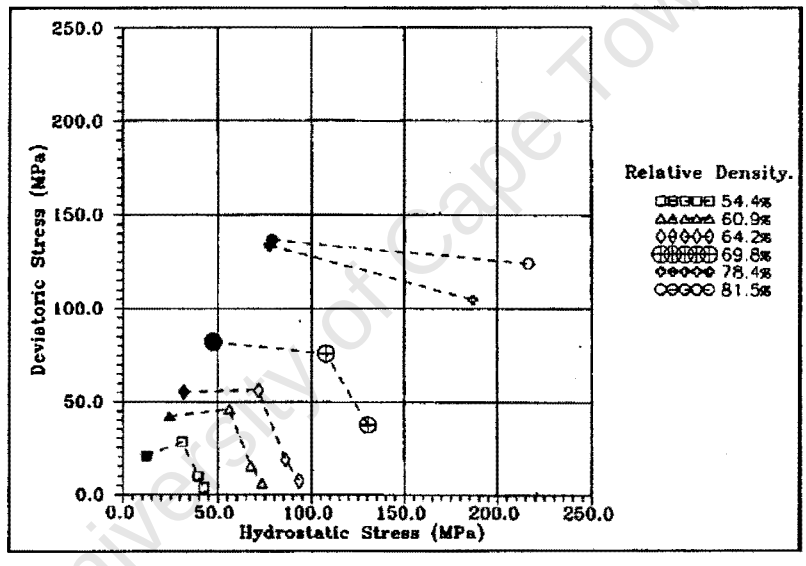

Figure 2.5: Brown's results for iron powder.

A number of variations to the quadratic yield surface have been proposed and they will be looked at briefly in the next section.

2.3.2 Variations of *Quadratic* Yield Surfaces

One variation of the *quadratic* yield surface is the yield surface model proposed by Gurson(1977:2). This model was developed for modelling ductile rupture in porous metals with some initial void ratio. It is basically a modification of the von Mises model that takes into account the effect that voids have on the failure of the material. Macroscopic compaction is possible, but the matrix material remains incompressible. The yield surface is therefore dependent on relative density and looks very similar to the quadratic yield surface of Figure 2.3. The form of the yield surface equation is

$$3\left(\frac{S}{\bar{\sigma}_y}\right)^2 + 2q_1(1-\rho)\cosh\left(q_2\frac{3\sigma_m}{2\bar{\sigma}_y}\right) - \left(1+q_3(1-\rho)^2\right) = 0 \qquad (2.8)$$

where S and σ_m are defined in Equations (2.5) and (2.6), ρ is the relative density, $\bar{\sigma}_y$ is the uniaxial yield strength of the matrix material as a function of plastic strain and q_1, q_2 and q_3 are material parameters.

Because the Gurson type yield surface has a similar shape to the *quadratic* yield surface the same limitations can be expected when modelling the compaction of some powders at low hydrostatic stress.

Fleck(1992:1139) proposed a yield surface model based on the yielding that occurs at the contacts between perfectly plastic spherical metal particles. The particles are bonded by isolated contacts and deformation is assumed to occur by plastic yielding of the material at or near these contacts. The yield surface for the model was developed from the yield surface that defines the plastic yielding at the contacts. The most noticeable difference between this model and a *quadratic* yield surface is the sharp vertex where the yield surface contacts the hydrostatic axis which allows deviatoric plastic strain in response to pure hydrostatic loading. Fleck indicated that, although these strains could occur locally between individual powder particles, the global yield surface may not show this behaviour.

The model developed by Fleck has not been compared much with experimental results (Watson, 1993:2072) but again, because it is similar to the *quadratic* surface in the region of the deviatoric stress axis, the same limitations are expected at low hydrostatic stresses for some powders.

The review of quadratic yield surfaces and other variations to the quadratic yield surface with similar characteristics shows that they may predict some of the characteristics of certain metal powders, but they do not predict the behaviour of all metal powders and therefore other yield surface forms must be considered.

2.3.3 Yield Surfaces Derived from Soil Plasticity Theory

The most noticeable difference between *quadratic* yield surfaces and yield surfaces derived from soil plasticity is the asymmetry that soil plasticity yield surfaces have with respect to hydrostatic tension and compression(Watson, 1993:2072). The cohesion between soil particles or metal powder particles is limited especially at the beginning of compaction and therefore this asymmetry makes sense.

Soil plasticity yield surfaces usually are constructed from a fixed Drucker-Prager failure surface that is a straight line in the deviatoric-hydrostatic stress space and can be represented by

$$S - \alpha\sigma_m - k = 0 \qquad (2.9)$$

where S and σ_m are defined in Equations (2.5) and (2.6) and α and k are material parameters. This failure surface is unbounded for large hydrostatic stresses and because of this no consolidation can occur under hydrostatic loading. The slope of the failure surface also prevents consolidation but allows plastic dilatation to occur. The latter phenomenon can be represented physically as the rolling or sliding of particles over each other and particle rearrangement that occurs at low hydrostatic stresses.

Consolidation is modelled in soil plasticity with the addition of a *hardening cap* (usually spherical or elliptical) which then bounds the failure surface for hydrostatic compression. It is expected that a metal powder would behave similarly to a soil during the initial stages of compaction and therefore using yield surfaces developed from soil plasticity is justified. For further consolidation in powder compaction it would seem desirable to have a model that evolves to a von Mises type surface.

Haggblad(1993:A5) investigated the performance of a *cap* model proposed by DiMaggio(1971:935). The model had a fixed failure surface and a *hardening cap* as shown in Figure 2.6. The failure surface can be seen to change from a Drucker-Prager surface at low hydrostatic stresses to the von Mises surface at high hydrostatic stresses. The equations for the failure surface and *cap* are

$$f_{failure} = S - \left(A - C\exp\left(B\sigma_m\right)\right) = 0 \qquad (2.10)$$

$$f_{cap} = S - \tfrac{1}{R}\sqrt{\left(X(\kappa) - L(\kappa)\right)^2 - \left(3\sigma_m - L(\kappa)\right)^2} = 0 \qquad (2.11)$$

where S and σ_m are defined in Equations (2.5) and (2.6), A, B and C are material parameters, κ is the hardening parameter for the cap, $X(\kappa)$ and $L(\kappa)$ are functions of the hardening parameter and R is a material parameter that defines the *cap* shape. For this particular model κ is a function of the volumetric plastic strain only.

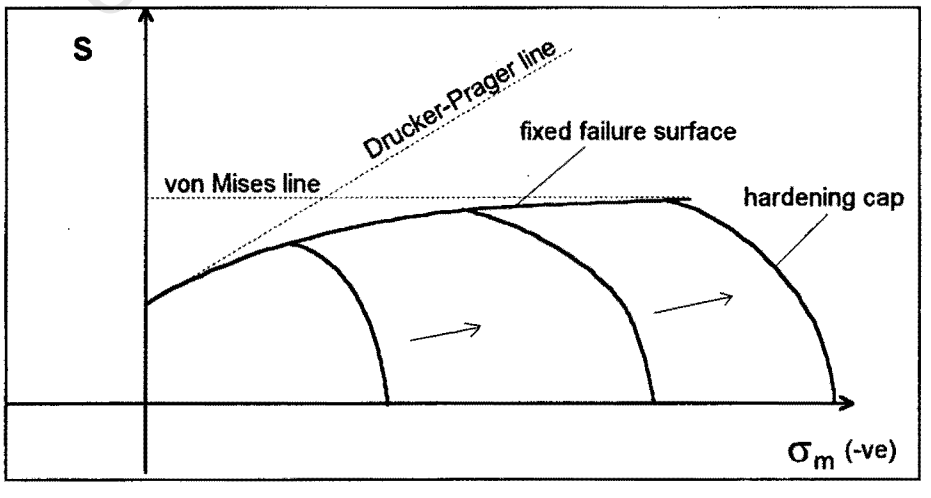

Figure 2.6: Soil plasticity type yield surface.

The limitation of this model is that it is difficult to calibrate using data from different loading setups (Haggblad, 1993:A4). The model was found to be easy to fit to hydrostatic loading data. The model therefore agreed well with experiments for hydrostatic loading but as the shear stress was increased deviation from the experimental results occurred. The advantage of this model is that it is relatively easy to calibrate from a few simple tests because there are only a few parameters.

Chtourou(1995:19) implemented the same *cap* model in an attempt to simulate the behaviour of stainless steel powder. Again the hardening for the *cap* was a function of volumetric plastic strain, ε^p_v, which was used to update the powder relative, ρ, as

$$\rho = \rho_0 e^{-\varepsilon_v^p} \qquad (2.12)$$

where ρ_0 is the initial relative density of the powder. Chtourou(1995:22) described the implementation of the constitutive equations in ABAQUS as a User-Material Subroutine and the experimental results from the compaction of a two level axisymmetric part were compared with finite element predicted results. The finite element results were found to agree well with the experimental results.

Watson(1993:2076) also proposed a *cap* yield surface for the simulation of aluminium powder compaction after extensive tests on aluminium powder. The shape of the yield surface was obtained by plotting isodensity data and fitting curves to the data. The data was obtained from constrained compression tests, hydrostatic tests and unconstrained compression and tension tests explained in Section 2.2. This data was plotted in the hydrostatic meridian plane which is the plane that contains the hydrostatic stress axis and the σ_3 principal stress axis. An example of data plotted for a particular relative density is shown in Figure 2.7.

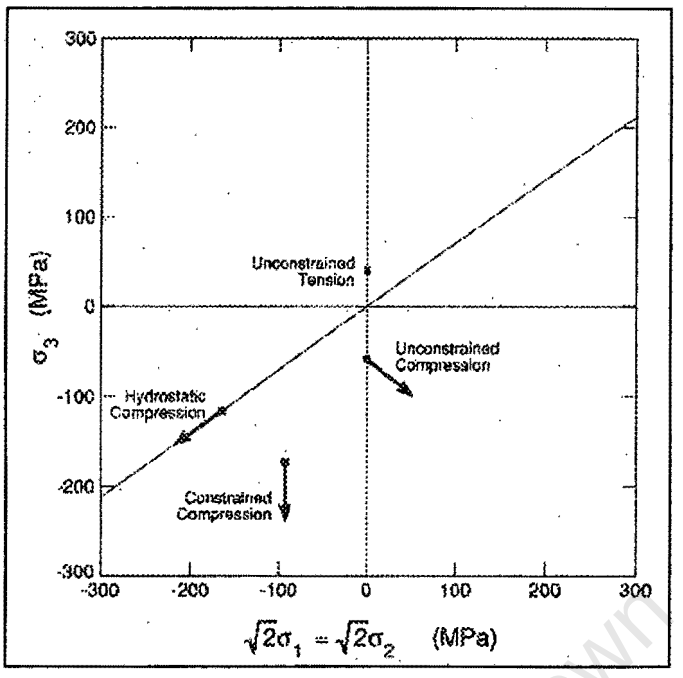

Figure 2.7: Watson's data for a particular relative density.

Watson(1993:2076) found that the curves that best fitted the data at a particular relative density were the combination of a Drucker-Prager yield surface which was named the shear yield surface and an elliptical *cap* which was named the consolidation yield surface. This model differed from the other *cap* models described as the shear surface was now not a fixed failure surface but a yield surface that evolved with density, similar to the consolidation yield surface, and approached the von Mises surface at full density.

The equations proposed by Watson(1993:2079) for the shear and consolidation yield surfaces involved the use of nine density dependent variables, as follows

$$f_{shear} = S + I_1\left(\frac{C_9}{C_7 + C_8}\right) - \left(\frac{C_7 C_9}{C_7 + C_8}\right) = 0 \qquad (2.13)$$

$$f_{cons} = S - C_3\left(1 - \frac{(I_1 - C_1)^2}{(C_2)^2}\right)^{\frac{1}{2}} = 0 \qquad (2.14)$$

where S is defined in Equation (2.6), I_1 is the first stress invariant($I_1=3\sigma_m$) and the C_i describe the yield surfaces at a fixed relative density. Watson(1993:2080) included data for the density dependent variables, C_i, for the full range of relative densities from 0.52 to full density. [See Appendix D].

The yield surfaces are represented graphically in Figure 2.8. It can be seen that the shear and consolidation yield surfaces intersect at some point along the consolidation yield surface which is not necessarily the apex of the elliptical *cap*. This was chosen so that the direction of plastic flow at this point, assuming an associative flow rule on both surfaces, was the same as that measured from the constrained compression tests. An associative flow rule on the shear yield surface also agreed with experimental observations. This, however, causes problems at the intersection of the two surfaces as the flow vector is not uniquely defined at that point.

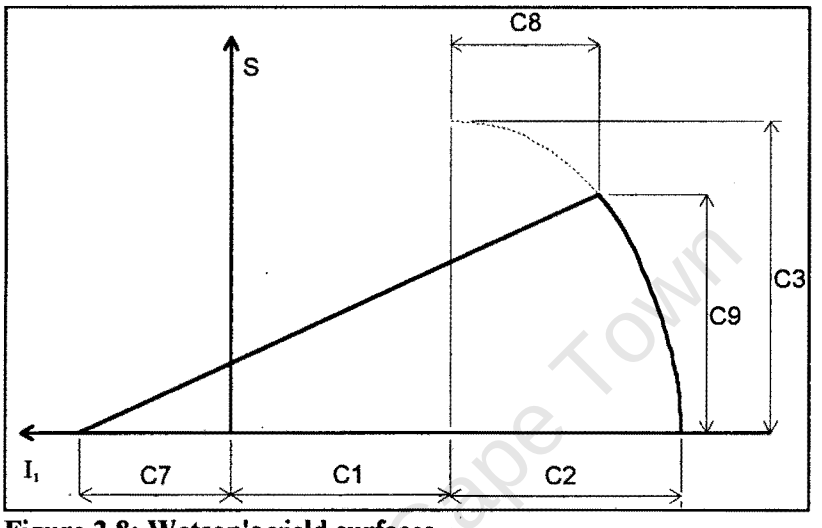

Figure 2.8: Watson's yield surfaces.

Watson(1993:2075) did not implement this model numerically but did outline some suggested constitutive equations for the model. A numerical analysis described by Sandler(1979:173) was suggested for numerical implementation of the constitutive equations as the analysis was used to implement equations with similar density hardening parameters.

An equation was suggested for the evolution of relative density, ρ, as a function of volumetric plastic strain, ε^p_v, as

$$\rho = \frac{\rho_0}{1+\varepsilon_v^p} \qquad (2.15)$$

where ρ_0 is the initial relative density of the powder. Hardening in a metal powder under consolidation can be considered to arise from three factors (Watson, 1993:2074). They are: an increase in relative density, work hardening in the vicinity of interparticle contacts due to volumetric plastic strain and work hardening in the vicinity of interparticle contacts due to deviatoric plastic strain. Therefore representing yielding as a function of relative density alone will neglect the contribution of deviatoric plastic strain in hardening. Watson(1993:2074) found, however, that for aluminium powder, hardening due to deviatoric plastic strain is negligible. Brown(1994:395) found that this was not the case for iron powder due to interparticle cohesion and therefore the

addition of another state variable based on cohesion may be needed to represent this behaviour.

The *cap* models which have been described in this section are more consistent with reported experimental work than the *quadratic* models for modelling metal powder compaction and the remainder of this thesis details the implementation of a *cap* model similar to that proposed by Watson(1993).

CHAPTER 3

A NEW CONSTITUTIVE MODEL FOR MODELLING METAL POWDER COMPACTION

After careful consideration of the work done by Watson(1993) it was decided to develop and implement a model similar to the one that was proposed. This decision was made based on a number of reasons. Firstly, Watson had done extensive tests on an actual metal powder using a full range of tests so that the mechanical behaviour of the aluminium powder was well understood. Secondly, the *cap* type yield surface that was proposed agreed well with the results that were obtained from experiment. Lastly, other sources (Brown, 1994) had found, through experiment, that *cap* type yield surfaces may be more appropriate for modelling powder compaction than *quadratic* yield surfaces. The rest of this chapter details the development of a new constitutive model based on the work done by Watson(1993) using a similar shear yield surface and consolidation yield surface combination but using a different approach to that suggested by Sandler(1979:173) for the numerical implementation of the constitutive equations.

3.1 YIELD SURFACE DEFINITION

In an attempt to simplify the yield surface equations proposed by Watson(1993:2079), the number of density dependent variables was reduced in the shear yield surface equation. The form of the two surfaces in terms of the invariants S and σ_m are

$$f_{shear} = S + \alpha_N(\rho)\sigma_m - k(\rho) = 0 \qquad (3.1)$$

$$f_{cons} = S^2 - \frac{(C_3(\rho))^2}{(C_2(\rho))^2}\left((C_2(\rho))^2 - \left(3\sigma_m - (C_1(\rho))\right)^2\right) = 0 \qquad (3.2)$$

where S and σ_m are defined in Equations (2.5) and (2.6), ρ is the relative density of the powder and $\alpha_N(\rho)$, $k(\rho)$ and $C_i(\rho)$ are density dependent variables. The variable, $\alpha_N(\rho)$, reduces to zero at full density so that the shear yield surface becomes a von Mises surface.

It can been seen that the form of the consolidation yield surface is different to that proposed by Watson. The most noticeable change is that the terms of the equation have been squared. This was done to remove the square root from the equation which

caused problems during numerical implementation. The shape of this equation in deviatoric-hydrostatic stress space, however, remains unchanged.

The yield surfaces are represented graphically in Figure 3.1 where it can be seen that the shear and consolidation yield surfaces intersect at some point along the consolidation yield surface. This point is assumed to be on the constrained compression deformation path although Watson proposed that the point was not necessarily at the apex of the consolidation yield surface. In this model this was simplified so that the tangent to the consolidation yield surface at the intersection of the two surfaces is horizontal. This condition was guaranteed by the relationship

$$C_3(\rho) = -\alpha_N(\rho)\frac{C_1(\rho)}{3} + k(\rho) \qquad (3.3)$$

where $C_1(\rho)$ is negative and $\alpha_N(\rho)$ is positive. This simplification was decided on as Watson's experimental data for aluminium, shown in Appendix D, indicated that this was the case for the aluminium powder. It cannot be guaranteed that this is true for all powders but the value of $C_3(\rho)$ can easily be adjusted by changing $k(\rho)$. This would probably not have a significant effect on the results as consolidation only occurs when yielding occurs on the consolidation yield surface.

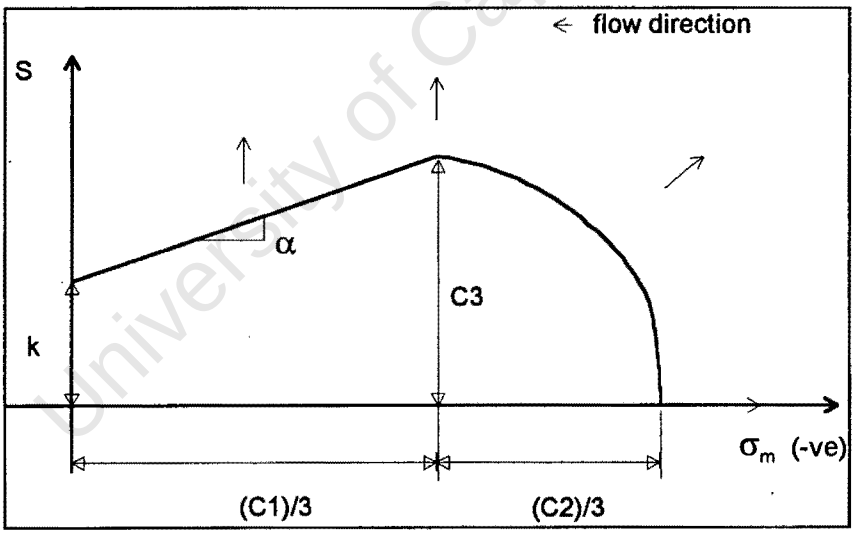

Figure 3.1: New model yield surfaces and defining variables.

Another problem that had to be addressed was the direction of plastic flow when yielding occurred on either of the two surfaces. Watson's(1993) experimental results indicated that associated flow on both surfaces would agree with experiment. If associative flow was assumed on both surfaces the plastic flow direction would not be uniquely defined at the intersection of the two surfaces and the plastic flow vector can assume any orientation between the normal vectors of the two intersecting surfaces which makes numerical implementation of the equations more difficult. Associative flow was assumed on the consolidation yield surface but not on the shear yield surface for this model. Plastic flow was defined in terms of a von Mises stress potential on the

shear yield surface to ensure the flow direction is continuous at the intersection of the two yield surfaces. The flow direction on the two surfaces is indicated in Figure 3.1. The simplification of adopting a von Mises flow potential on the shear yield surface will have consequences on the operation of the model in the shear region. The physical behaviour of particle rearrangement at low hydrostatic stresses that is manifested as plastic dilatation with an associative flow rule is lost and is represented as yielding without change in volumetric plastic strain with a von Mises flow potential. This prevents 'softening' on the consolidation yield surface in response to yielding on the shear yield surface that occurs with an associative flow rule as no increase in volumetric plastic strain occurs. These limitations, however, have to be looked at in the context of metal powder compaction where the applied stresses are very high and particle rearrangement at low hydrostatic stresses is not significant against consolidation which occurs on the consolidation yield surface. The simplifications made are therefore justified for the application of metal powder compaction. Future work could involve the investigation of the effect of different flow potentials on the performance of the shear yield surface and how this affects the coupling between the shear and consolidation yield surfaces.

The condition for determining which of the two yield surfaces may be active is dependent on the value of the hydrostatic stress

$$\sigma_m > \frac{C_1(\rho)}{3} \Rightarrow \text{shear yield surface}$$

$$\sigma_m \leq \frac{C_1(\rho)}{3} \Rightarrow \text{consolidation yield surface} \qquad (3.4)$$

If yielding occurs a change of relative density may occur. This change can be related to the incremental change in volume as follows (Aravas, 1987:1406)

$$d\rho = -\rho d\varepsilon_v^p \qquad (3.5)$$

where $d\rho$ is the change in relative density and $d\varepsilon^p_v$ is the change is volumetric plastic strain. This equation can be integrated to give

$$\rho = \rho_0 e^{-\varepsilon_v^p} \qquad (3.6)$$

where ρ_0 is the initial relative density of the powder. The relative density is updated in this model using this equation. This is different to the equation that was proposed by Watson(1993:2074). The integration of Equation (3.5) and a comparison of Watson's Equation (2.15) and Equation (3.6) are given in Appendix A.

3.2 PLASTICITY CONSTITUTIVE EQUATIONS

The constitutive equations were developed using a method similar to that used by Resende(1985:855) for the development of a *cap* model. The basic plasticity constitutive equations will first be developed and these will then be applied to the shear and consolidation yield surfaces. Isotropic behaviour is assumed.

3.2.1 Basic Plasticity Rate Equations

It is assumed that the total strain rate $\dot{\varepsilon}_{ij}$ can be written as the sum of elastic and plastic components as

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}^{e}_{ij} + \dot{\varepsilon}^{p}_{ij} \qquad (3.7)$$

The elastic strains can be split into volumetric and deviatoric parts so that the elastic constitutive equations can be written as

$$\dot{\varepsilon}^{e}_{kk} = \frac{\dot{\sigma}_{kk}}{3K(\rho)}; \quad \dot{e}^{e}_{ij} = \frac{\dot{S}_{ij}}{2G(\rho)} \qquad (3.8)$$

where $K(\rho)$ and $G(\rho)$ are the density dependent bulk and shear moduli, respectively, and $\dot{e}_{ij}$ and $\dot{S}_{ij}$ are the deviatoric components of $\dot{\varepsilon}_{ij}$ and $\dot{\sigma}_{ij}$, respectively. The volumetric strain rate can be rewritten as

$$\dot{\varepsilon}^{e}_{kk} = \frac{\dot{\sigma}_{m}}{K(\rho)} \qquad (3.9)$$

where $\dot{\sigma}_{m} = \frac{1}{3}\dot{\sigma}_{kk}$ is the hydrostatic stress rate.

The plastic strain rate $\dot{\varepsilon}^{p}_{ij}$ is assumed to be proportional to the stress gradient of a function termed the plastic potential and can be written as

$$\dot{\varepsilon}^{p}_{ij} = \lambda_{\alpha} \frac{\partial g_{\alpha}}{\partial \sigma_{ij}} \qquad (3.10)$$

where α refers to either the shear or consolidation surface, λ_{α} is the plastic multiplier and g_{α} is the plastic potential. For associative flow $g_{\alpha}=f_{\alpha}$. The conditions for plastic

loading or unloading at a material point in terms of the yield function and plastic multiplier can be written as (Lubliner, 1990)

$$\begin{aligned} & f_\alpha \le 0 \\ & \lambda_\alpha \ge 0 \\ \text{and } & f_\alpha \lambda_\alpha = 0 \end{aligned} \qquad (3.11)$$

Since the yield surfaces were written in terms of the two invariants S and σ_m, it is convenient to rewrite Equation (3.10) as

$$\dot{\varepsilon}^p{}_{ij} = \lambda_\alpha \left(\frac{\partial g_\alpha}{\partial \sigma_m} \frac{\partial \sigma_m}{\partial \sigma_{ij}} + \frac{\partial g_\alpha}{\partial S} \frac{\partial S}{\partial \sigma_{ij}} \right) \qquad (3.12)$$

where

$$\frac{\partial \sigma_m}{\partial \sigma_{ij}} = \frac{1}{3}\delta_{ij}, \quad \frac{\partial S}{\partial \sigma_{ij}} = \frac{S_{ij}}{2S} \qquad (3.13)$$

Hence, the volumetric plastic strain rate can be written as

$$\dot{\varepsilon}^p_{kk} = \lambda_\alpha \frac{\partial g_\alpha}{\partial \sigma_m} \qquad (3.14)$$

and the deviatoric plastic strain rate can be written as

$$\dot{e}^p_{ij} = \frac{\lambda_\alpha}{2S} \frac{\partial g_\alpha}{\partial S} S_{ij} \qquad (3.15)$$

These equations can be simplified so that they can be represented in the two dimensional stress space of the invariants σ_m and S. The effective volumetric strain rate can simply be defined as

$$\dot{\varepsilon}_v = \dot{\varepsilon}_{kk} \qquad (3.16)$$

An effective shear strain rate can be defined as (Resende, 1985:857)

$$\dot{e} = \frac{1}{S} S_{ij} \dot{e}_{ij} \qquad (3.17)$$

and the shear stress rate $\dot{S}$ can be obtained by differentiating Equation (2.6) and is written as

$$\dot{S} = \frac{1}{2S} S_{ij} \dot{S}_{ij} \qquad (3.18)$$

These invariant relationships can now be used to rewrite the constitutive equations. The effective strain rates can be written as

$$\dot{e} = \dot{e}^e + \dot{e}^p \; ; \qquad \dot{\varepsilon}_v = \dot{\varepsilon}_v^e + \dot{\varepsilon}_v^p \qquad (3.19)$$

and the elastic relations as

$$\dot{\varepsilon}_v^e = \frac{\dot{\sigma}_m}{K(\rho)}; \qquad \dot{e}^e = \frac{\dot{S}}{G(\rho)} \qquad (3.20)$$

and the plastic relations as

$$\dot{\varepsilon}_v^p = \lambda_\alpha \frac{\partial g_\alpha}{\partial \sigma_m}; \qquad \dot{e}^p = \lambda_\alpha \frac{\partial g_\alpha}{\partial S} \qquad (3.21)$$

3.2.2 Shear Yield Surface Equations

The shear yield surface equation as given in Equation (3.1) is

$$f_{shear} = S + \alpha_N(\rho)\sigma_m - k(\rho) = 0$$

The plastic potential is

$$g_{shear} = S \qquad (3.22)$$

If yielding occurs on the shear yield surface $f_{shear} = 0$, $\sigma_m > \frac{C_1(\rho)}{3}$. The effective plastic strain rates can be written as

$$\dot{e}^p = \lambda_{shear} \frac{\partial g_{shear}}{\partial S} = \lambda_{shear} \qquad (3.23)$$

$$\dot{\varepsilon}_v^p = \lambda_{shear} \frac{\partial g_{shear}}{\partial \sigma_m} = 0 \qquad (3.24)$$

The total strain rates are therefore

$$\dot{\varepsilon}_v = \frac{\dot{\sigma}_m}{K(\rho)}; \qquad \dot{e} = \frac{\dot{S}}{G(\rho)} + \lambda_{shear} \tag{3.25}$$

and the stress rates become

$$\dot{\sigma}_m = K(\rho)(\dot{\varepsilon}_v), \qquad \dot{S} = G(\rho)(\dot{e} - \lambda_{shear}) \tag{3.26}$$

3.2.3 Consolidation Yield Surface Equations

The consolidation yield surface equation as given in Equation (3.2) is

$$f_{cons} = S^2 - \frac{(C_3(\rho))^2}{(C_2(\rho))^2}\left((C_2(\rho))^2 - (3\sigma_m - (C_1(\rho)))^2\right) = 0$$

Assuming associative flow on this surface, the plastic potential is

$$g_{cons} = f_{cons} \tag{3.27}$$

If yielding occurs on the consolidation yield surface $f_{cons} = 0$, $\sigma_m \leq \frac{C_1(\rho)}{3}$. The effective plastic strain rates can be written as

$$\dot{e}^p = \lambda_{cons}\frac{\partial f_{cons}}{\partial S} = 2S\lambda_{cons} \tag{3.28}$$

$$\dot{\varepsilon}_v^p = \lambda_{cons}\frac{\partial f_{cons}}{\partial \sigma_m} \tag{3.29}$$

The total strain rates are therefore

$$\dot{\varepsilon}_v = \frac{\dot{\sigma}_m}{K(\rho)} + \lambda_{cons}\frac{\partial f_{cons}}{\partial \sigma_m}; \qquad \dot{e} = \frac{\dot{S}}{G(\rho)} + 2S\lambda_{cons} \tag{3.30}$$

and the stress rates become

$$\dot{\sigma}_m = K(\rho)\left(\dot{\varepsilon}_v - \lambda_{cons}\frac{\partial f_{cons}}{\partial \sigma_m}\right), \qquad \dot{S} = G(\rho)(\dot{e} - 2S\lambda_{cons}) \tag{3.31}$$

That completes the development of the constitutive rate equations for the new model.

CHAPTER 4

IMPLEMENTATION OF NEW CONSTITUTIVE MODEL

The rate equations of the constitutive model presented in Chapter Three must be recast in an incremental form for implementation into a finite element program. The essential features of an incremental formulation are that given the solution at increment n-1 (σ_{n-1}, ε_{n-1}, ε^p_{n-1}, ρ_{n-1}) the new values σ_n, ε^p_n, ρ_n must be calculated. This means that given the stress from the previous increment, σ_{n-1}, and a new increment in strain, $\Delta\varepsilon_n$, the constitutive model must calculate the resultant stress, σ_n, the increment in plastic strain, $\Delta\varepsilon^p_n$, and the consistent modulus. The development of the incremental stress update procedures for the shear and consolidation yield surfaces and their subsequent numerical implementation are discussed in this chapter.

4.1 STRESS UPDATE CALCULATIONS

The stress update and calculation of the plastic strain increment is done using an elastic predictor-plastic corrector method. This method calculates an elastic predictor, assuming that the increment in strain, $\Delta\varepsilon_n$, is initially elastic. If the stress state calculated is elastic then the assumption was correct and the final stress is determined. If, however, the elastic predictor falls outside the yield surface, a plastic corrector step must follow to return the stress point onto the yield surface, calculate the increment in plastic strain and update other quantities.

The implementation takes advantage of the invariant form of the deviatoric and volumetric elastic-plastic equations as developed in Chapter Three. Thus the incremental formulation of Equations (3.19), (3.20) and (3.21) using an Euler Backward integration scheme can be written as

$$\Delta e_n = \Delta e^e_n + \Delta e^p_n \,, \qquad (\Delta\varepsilon_v)_n = (\Delta\varepsilon^e_v)_n + (\Delta\varepsilon^p_v)_n \tag{4.1}$$

$$\begin{aligned} \Delta S_n &= G(\rho_n)(\Delta e_n - \Delta e^p_n) \\ (\Delta\sigma_m)_n &= K(\rho_n)((\Delta\varepsilon_v)_n - (\Delta\varepsilon^p_v)_n) \end{aligned} \tag{4.2}$$

$$\Delta e_n^p = \lambda_\alpha \left.\frac{\partial g_\alpha}{\partial S}\right|_n$$
$$\left(\Delta \varepsilon_v^p\right)_n = \lambda_\alpha \left.\frac{\partial g_\alpha}{\partial \sigma_m}\right|_n \qquad (4.3)$$

The updated quantities at the end of the increment n are

$$e_n = e_{n-1} + \Delta e_n, \quad (\varepsilon_v)_n = (\varepsilon_v)_{n-1} + (\Delta \varepsilon_v)_n \qquad (4.4)$$

$$e_n^p = e_{n-1}^p + \Delta e_n^p, \quad \left(\varepsilon_v^p\right)_n = \left(\varepsilon_v^p\right)_{n-1} + \left(\Delta \varepsilon_v^p\right)_n \qquad (4.5)$$

$$S_n = S_{n-1} + \Delta S_n, \quad (\sigma_m)_n = (\sigma_m)_{n-1} + (\Delta \sigma_m)_n \qquad (4.6)$$

Substituting (4.2) into (4.6) gives

$$S_n = S_{n-1} + G(\rho_n)\left(\Delta e_n - \Delta e_n^p\right)$$
$$(\sigma_m)_n = (\sigma_m)_{n-1} + K(\rho_n)\left((\Delta \varepsilon_v)_n - \left(\Delta \varepsilon_v^p\right)_n\right) \qquad (4.7)$$

To simplify the stress update algorithm it is now assumed that $G(\rho)$ and $K(\rho)$ are evaluated using ρ_{n-1} instead of ρ_n. This simplification is justified in this case as the constitutive behaviour of metal powder has a weak dependence on relative density (Brown, 1994:385). This was confirmed by a number of numerical tests. Since Δe_n and $(\Delta \varepsilon_v)_n$ are known from the global finite element solution the above equations can be written in the conventional form

$$S_n = S_n^E - G(\rho_{n-1})\Delta e_n^p$$
$$(\sigma_m)_n = \left(\sigma_m^E\right)_n - K(\rho_{n-1})\left(\Delta \varepsilon_v^p\right)_n \qquad (4.8)$$

where S^E and σ_m^E are the elastic predictor stresses. Substituting Equations (4.3) into (4.8) gives

$$S_n = S_n^E - G(\rho_{n-1})\lambda_\alpha \left.\frac{\partial g_\alpha}{\partial S}\right|_n$$
$$(\sigma_m)_n = (\sigma_m^E)_n - K(\rho_{n-1})\lambda_\alpha \left.\frac{\partial g_\alpha}{\partial \sigma_m}\right|_n \quad (4.9)$$

The plastic multiplier, λ_α, is solved for by satisfying the plastic consistency condition

$$f_\alpha\left(S_n, (\sigma_m)_n, \kappa(\rho_n)\right) = 0 \quad (4.10)$$

where $\kappa(\rho_n)$ represents the density dependent variables which determine the shapes of the two yield surfaces. By substituting for S_n, $(\sigma_m)_n$ and $\kappa(\rho_n)$ into the shear and consolidation yield surfaces, (4.10) can be written as a function of λ_α as

$$f_\alpha(\lambda_\alpha) = 0 \quad (4.11)$$

In general this is a non-linear equation which can be solved by means of a Newton-Raphson scheme as

$$\lambda_\alpha^{i+1} = \lambda_\alpha^i - \frac{f_\alpha(\lambda_\alpha^i)}{f_\alpha'(\lambda_\alpha^i)} \quad (4.12)$$

where $f_\alpha'(\lambda_\alpha)$ indicates differentiation with respect to λ_α.

The update procedure can now be summarised as follows:

Initialise variables:

$i = 0$

$(\,)_n^0 = (\,)_{n-1}$

$\lambda_\alpha^0 = 0$

1. $f_\alpha(\lambda_\alpha^i) \le 0 \Rightarrow$ stresses elastic or on yield surface
2. $f_\alpha(\lambda_\alpha^i) > 0$
   - solve for λ_α^{i+1} from (4.12)
   - update stresses and state variables

$$S_n^{i+1} = S_n^E - G(\rho_{n-1})\lambda_\alpha^{i+1}\left.\frac{\partial g_\alpha}{\partial S}\right|_n^i$$
$$(\sigma_m)_n^{i+1} = (\sigma_m^E)_n - K(\rho_{n-1})\lambda_\alpha^{i+1}\left.\frac{\partial g_\alpha}{\partial \sigma_m}\right|_n^i$$
$$(\varepsilon_v^p)_n^{i+1} = (\varepsilon_v^p)_{n-1} + \lambda_\alpha^{i+1}\left.\frac{\partial g_\alpha}{\partial \sigma_m}\right|_n^i$$
$$\rho_n^{i+1} = \rho_0 e^{(\varepsilon_v^p)_n^{i+1}}$$

3. i=i+1, go to 1.

4.1.1 Shear Yield Surface Stress Update Calculations

For the shear yield surface

$$\Delta e_n^p = \lambda_{shear} \tag{4.13}$$

$$(\Delta\varepsilon_v^p)_n = 0 \tag{4.14}$$

Equations (4.8) become

$$S_n = S_n^E - G(\rho_{n-1})\lambda_{shear}$$
$$(\sigma_m)_n = (\sigma_m^E)_n \tag{4.15}$$

and λ_{shear} can be solved for directly from (4.12) as

$$\lambda_{shear} = \frac{(f_{shear})^E}{G(\rho_{n-1})} \tag{4.16}$$

The full derivation of Equation (4.16) is shown in Appendix C. No iteration is needed in this case.

To obtain the components of the deviatoric stress tensor note that

$$\begin{aligned} S_{ij_n} &= S_{ij_{n-1}} + \Delta S_{ij_n} \\ &= S_{ij_{n-1}} + 2G(\rho_{n-1})(\Delta e_{ij_n}) - 2G(\rho_{n-1})(\Delta e_{ij_n}^p) \\ &= S_{ij_n}^E - 2G(\rho_{n-1})(\Delta e_{ij_n}^p) \end{aligned} \tag{4.17}$$

From (3.15) the plastic strain increment is

$$\Delta e^{p}_{ij_n} = \frac{\lambda_{shear}}{2S_n} S_{ij_n} \qquad (4.18)$$

Substituting Equation (4.18) into Equation (4.17) and rearranging gives the stress update for the deviatoric stress components as

$$S_{ij_n} = \frac{S^{E}_{ij_n}}{1 + \frac{G(\rho_{n-1})\lambda_{shear}}{S_n}} \qquad (4.19)$$

4.1.2 Consolidation Yield Surface Stress Update Equations

For the consolidation yield surface

$$\Delta e^{p}_{n} = 2S_n \lambda_{cons} \qquad (4.20)$$

$$\left(\Delta \varepsilon^{p}_{v}\right)_n = \lambda_{cons} \left.\frac{\partial f_{cons}}{\partial \sigma_m}\right|_n = \lambda_{cons} N\left(\rho_n, (\sigma_m)_n\right) \qquad (4.21)$$

where $N(\rho_n,(\sigma_m)_n)$ is the derivative $\left.\frac{\partial f_{cons}}{\partial \sigma_m}\right|_n$ [see Appendix B].

Equations (4.8) become

$$\begin{aligned} S_n &= S^{E}_{n} - 2G(\rho_{n-1})S_n \lambda_{cons} \\ (\sigma_m)_n &= \left(\sigma^{E}_{m}\right)_n - K(\rho_{n-1})\lambda_{cons} N\left(\rho_n, (\sigma_m)_n\right) \end{aligned} \qquad (4.22)$$

In this case, an iterative stress return is necessary to solve for the plastic multiplier, λ_{cons}. Equation (4.12) becomes

$$\lambda^{i+1}_{cons} = \lambda^{i}_{cons} - \frac{f\left(\lambda^{i}_{cons}\right)_{cons}}{f'\left(\lambda^{i}_{cons}\right)_{cons}} \qquad (4.23)$$

The full derivation and solution of Equation (4.23) are shown in Appendix C.

The deviatoric stress invariant is therefore updated according to

$$S_n^{i+1} = S_n^E - 2G(\rho_{n-1})S_n^{i+1}\lambda_{cons}^i \quad (4.24)$$

which can be rewritten as

$$S_n^{i+1} = \frac{S_n^E}{1 + 2G(\rho_{n-1})\lambda_{cons}^i} \quad (4.25)$$

The components of the deviatoric stress tensor were obtained using Equation (4.17) and substituting for the increment in plastic strain with

$$\Delta e_{ij_n}^p = \lambda_{cons} S_{ij_n} \quad (4.26)$$

derived from (3.15). After simplification the stress update for the deviatoric components becomes

$$S_{ij_n} = \frac{S_{ij_n}^E}{1 + 2G(\rho_{n-1})\lambda_{cons}} \quad (4.27)$$

The hydrostatic stress update requires an additional internal iteration loop to achieve convergence of the solution. The hydrostatic stress was updated according to

$$\left(\left(\varepsilon_v^p\right)_n^{i+1}\right)^{k+1} = \left(\varepsilon_v^p\right)_{n-1} + N\left(\left(\rho_n^i\right)^k, \left(\left(\sigma_m\right)_n^i\right)^k\right)\lambda_{cons}^{i+1} \quad (4.28)$$

$$\left(\rho_n^{i+1}\right)^{k+1} = \rho_0 e^{-\left(\left(\varepsilon_v^p\right)_n^{i+1}\right)^{k+1}} \quad (4.29)$$

$$\left(\left(\sigma_m\right)_n^{i+1}\right)^{k+1} = \left(\sigma_m^E\right)_n - K(\rho_{n-1})N\left(\left(\rho_n^{i+1}\right)^{k+1}, \left(\left(\sigma_m\right)_n^i\right)^k\right)\lambda_{cons}^{i+1} \quad (4.30)$$

The update algorithm described above performed similarly and, in some cases, better than the update scheme proposed by Ortiz(1986:353).

4.2 CONSISTENT MODULI

The consistent modulus can generally be referred to as the *slope* of the stress-strain curve at a certain strain point. It is required when a Newton-Raphson scheme is used to solve the global Finite Element System of equations.

The consistent moduli for the two yield surfaces are derived in the same manner as the constitutive equations, by working with the deviatoric parts and volumetric parts separately. The derivation was done in tensor notation and was reduced to vector notation for numerical implementation.

4.2.1 Shear Yield Surface Consistent Modulus

Using an Euler Backward scheme the increment in deviatoric stress at any increment n can be written as

$$\left(\Delta S_{ij}\right)_n = 2G\left(\rho_{n-1}\right)\left(\Delta e_{ij}\right)_n - 2G\left(\rho_{n-1}\right)\left(\Delta e^p_{ij}\right)_n \qquad (4.31)$$

Differentiating Equation (4.31) and discarding the increment notation gives

$$dS_{ij} = 2Gde_{ij} - 2Gde^p_{ij} \qquad (4.32)$$

where $G = G(\rho_{n-1})$ for this derivation. This can be expanded by substituting for de^p_{ij} with

$$de^p_{ij} = d\lambda_{shear}\frac{S_{ij}}{2S} + \lambda_{shear}\frac{dS_{ij}}{2S} - \lambda_{shear}\frac{S_{ij}}{2S^2}dS \qquad (4.33)$$

Therefore Equation (4.32) becomes

$$\begin{aligned} dS_{ij} &= 2Gde_{ij} - Gd\lambda_{shear}\frac{S_{ij}}{S} - G\lambda_{shear}\frac{dS_{ij}}{S} + G\lambda_{shear}\frac{S_{ij}}{S^2}dS \\ &= \left(1+\frac{G\lambda_{shear}}{S}\right)^{-1}\left[2Gde_{ij} - Gd\lambda_{shear}\frac{S_{ij}}{S} + G\lambda_{shear}\frac{S_{ij}}{S^2}dS\right] \end{aligned} \qquad (4.34)$$

Both dS and $d\lambda$ must be substituted for. From (4.2) dS can be written as

$$dS = Gde - Gd\lambda_{shear} \qquad (4.35)$$

The shear yield surface equation can be differentiated to give

$$df_{shear} = \frac{\partial f_{shear}}{\partial S} dS + \frac{\partial f_{shear}}{\partial \sigma_m} d\sigma_m + \frac{\partial f_{shear}}{\partial \alpha_N} d\alpha_N + \frac{\partial f_{shear}}{\partial k} dk = 0 \qquad (4.36)$$

where $d\alpha$ and dk are zero, dS is defined in Equation (4.35) and $d\sigma_m$ is defined by

$$d\sigma_m = K d\varepsilon_v \qquad (4.37)$$

where $K = K(\rho)$ for this derivation. Substituting (4.35) and (4.37) into (4.36) gives

$$df_{shear} = Gde - Gd\lambda_{shear} + \alpha_N K d\varepsilon_v = 0 \qquad (4.38)$$

$$d\lambda_{shear} = \frac{Gde + \alpha_N K d\varepsilon_v}{G} \qquad (4.39)$$

Substituting (4.35) and (4.39) into (4.34) and simplifying gives

$$dS_{ij} = \left(1 + \frac{G\lambda_{shear}}{S}\right)^{-1} \left[2Gde_{ij} + \left(\frac{G^2 \lambda_{shear} S_{ij}}{S^2} - \left(\frac{GS_{ij}}{S} + \frac{G^2 \lambda_{shear} S_{ij}}{S^2} \right) \right) de - \frac{\alpha_N K}{G} \left(\frac{GS_{ij}}{S} + \frac{G^2 \lambda_{shear} S_{ij}}{S^2} \right) d\varepsilon_v \right] \qquad ...(4.40)$$

From (3.17)

$$de = \frac{1}{S} S_{kl} de_{kl} \qquad (4.41)$$

Substituting (4.41) and simplifying gives

$$dS_{ij} = \left[\left(\frac{2G}{\left(1 + \frac{G\lambda_{shear}}{S}\right)} \delta_{ik}\delta_{jl} + \left(\frac{G^2\lambda_{shear}}{S^3\left(1 + \frac{G\lambda_{shear}}{S}\right)} - \frac{G}{S^2} \right) S_{ij}S_{kl} \right) de_{kl} - \frac{\alpha_N K S_{ij}}{S} d\varepsilon_v \right] \quad \text{...(4.42)}$$

The increment in volumetric stress at any increment n is given by (4.37). Equations (4.37) and (4.42) can now be combined to give

$$\begin{bmatrix} \frac{dS_{ij}}{d\sigma_m} \end{bmatrix} = \left[\begin{array}{c|c} A_{ijkl} & B_{ij} \\ \hline C_{kl} & D \end{array} \right] \begin{bmatrix} \frac{de_{kl}}{d\varepsilon_v} \end{bmatrix} \quad \text{(4.43)}$$

where $C_{kl} = 0$ and $D = K$.

4.2.2 Consolidation Yield Surface Consistent Modulus

The derivation starts as with the shear yield surface from

$$dS_{ij} = 2Gde_{ij} - 2Gde^p_{ij} \quad \text{(4.44)}$$

where $G = G(\rho_{n-1})$ for this derivation. This can be expanded by substituting for de^p_{ij} with

$$de^p_{ij} = d\lambda_{cons}S_{ij} + \lambda_{cons}dS_{ij} \quad \text{(4.45)}$$

Therefore Equation (4.44) becomes

$$\begin{aligned} dS_{ij} &= 2Gde_{ij} - 2Gd\lambda_{cons}S_{ij} - 2G\lambda_{cons}dS_{ij} \\ &= \left(1 + 2G\lambda_{cons}\right)^{-1}\left[2Gde_{ij} - 2Gd\lambda_{cons}S_{ij}\right] \end{aligned} \quad \text{(4.46)}$$

The derivation of $d\lambda$ is as follows. The consolidation yield surface equation can be differentiated to give

$$df_{cons} = \frac{\partial f_{cons}}{\partial S} dS + \frac{\partial f_{cons}}{\partial \sigma_m} d\sigma_m + \frac{\partial f_{cons}}{\partial C_1} dC_1 + \frac{\partial f_{cons}}{\partial C_2} dC_2 + \frac{\partial f_{cons}}{\partial C_3} dC_3 = 0 \quad \text{...(4.47)}$$

where

$$dS = \left(1 + 2G\lambda_{cons}\right)^{-1}\left(Gde - 2GSd\lambda_{cons}\right) \quad (4.48)$$

and

$$\begin{aligned} d\sigma_m &= Kd\varepsilon_v - Kd\varepsilon_v^p \\ &= Kd\varepsilon_v - Kd\lambda_{cons}N(\rho,\sigma_m) - K\lambda_{cons}T(\rho)d\sigma_m \end{aligned} \quad (4.49)$$

where $K = K(\rho)$ for this derivation.

$$d\sigma_m = \left(1 + K\lambda_{cons}T(\rho)\right)^{-1}\left(Kd\varepsilon_v - Kd\lambda_{cons}N(\rho,\sigma_m)\right) \quad (4.50)$$

and

$$\begin{aligned} dC_1 &= \frac{\partial C_1}{\partial \rho}\frac{\partial \rho}{\partial \varepsilon_v^p} d\varepsilon_v^p \\ &= Ed\varepsilon_v^p \\ &= Ed\lambda_{cons}N(\rho,\sigma_m) - E\lambda_{cons}T(\rho)d\sigma_m \end{aligned} \quad (4.51)$$

$$\begin{aligned} dC_2 &= \frac{\partial C_2}{\partial \rho}\frac{\partial \rho}{\partial \varepsilon_v^p} d\varepsilon_v^p \\ &= Fd\varepsilon_v^p \\ &= Fd\lambda_{cons}N(\rho,\sigma_m) - F\lambda_{cons}T(\rho)d\sigma_m \end{aligned} \quad (4.52)$$

$$\begin{aligned} dC_3 &= \frac{\partial C_3}{\partial \rho}\frac{\partial \rho}{\partial \varepsilon_v^p} d\varepsilon_v^p \\ &= Md\varepsilon_v^p \\ &= Md\lambda_{cons}N(\rho,\sigma_m) - M\lambda_{cons}T(\rho)d\sigma_m \end{aligned} \quad (4.53)$$

where $N(\rho,\sigma_m)$ and $T(\rho)$ are defined in Appendix B. Substituting (4.48) and (4.50)-(4.53) into (4.47) and solving for $d\lambda$ gives

$$d\lambda_{cons} = \frac{\dfrac{2SG}{\left(1+2G\lambda_{cons}\right)}de + XKd\varepsilon_v}{C_C} \tag{4.54}$$

where

$$X = \frac{\left(N(\rho,\sigma_m) + PE\lambda_{cons}T(\rho) + QF\lambda_{cons}T(\rho) + RM\lambda_{cons}T(\rho)\right)}{\left(1+K\lambda_{cons}T(\rho)\right)} \tag{4.55}$$

and

$$C_C = \frac{4S^2G}{\left(1+2G\lambda_{cons}\right)} - N(\rho,\sigma_m)(PE+QF+RM) + XN(\rho,\sigma_m)K \tag{4.56}$$

and P, Q and R are defined in Appendix B.

Substituting (4.38) and (4.54) into (4.46) and simplifying gives

$$dS_{ij} = \left(1+2G\lambda_{cons}\right)^{-1}\left[\left(2G\delta_{ik}\delta_{jl} - \left(\frac{4G^2}{C_C\left(1+2G\lambda_{cons}\right)}\right)S_{ij}S_{kl}\right)de_{kl} - \frac{2GXK}{C_C}S_{ij}d\varepsilon_v\right] \tag{4.57}$$

A similar procedure can be used for the hydrostatic stress. Starting from (4.50)

$$d\sigma_m = \left(1+K\lambda_{cons}T(\rho)\right)^{-1}\left(Kd\varepsilon_v - Kd\lambda_{cons}N(\rho,\sigma_m)\right) \tag{4.58}$$

and substituting from (4.38) and (4.54) and simplifying gives

$$d\sigma_m = \frac{-2KN(\rho,\sigma_m)G}{C_c\left(1+K\lambda_{cons}T(\rho)\right)\left(1+2G\lambda_{cons}\right)}S_{kl}de_{kl} + \frac{\left(K-\dfrac{K^2N(\rho,\sigma_m)X}{C_c}\right)}{\left(1+K\lambda_{cons}T(\rho)\right)}d\varepsilon_v \qquad (4.59)$$

Equations (4.57) and (4.59) can now be written in the same form as (4.43) as

$$\begin{bmatrix} \dfrac{dS_{ij}}{d\sigma_m} \end{bmatrix} = \left[\begin{array}{c|c} A_{ijkl} & B_{ij} \\ \hline C_{kl} & D \end{array}\right] \begin{bmatrix} \dfrac{de_{kl}}{d\varepsilon_v} \end{bmatrix}$$

4.3 FINITE ELEMENT IMPLEMENTATION OF CONSTITUTIVE EQUATIONS

The stress update and consistent moduli calculations discussed in the previous sections must be carried out at each material point in a finite element mesh. This has been done by implementing the incremental formulation into a User-Material Subroutine in ABAQUS. The resultant FORTRAN 77 code is listed in Appendix F. The User-Material Subroutine can handle both three dimensional and axisymmetric problems. Only the three dimensional case will be discussed here.

The stress and strain vectors which are used in the finite element implementation for a three dimensional stress state are written as

$$\underline{\sigma} = \left(\sigma_{11}\ \sigma_{22}\ \sigma_{33}\ \sigma_{12}\ \sigma_{13}\ \sigma_{23}\right)^T \qquad (4.60)$$

$$\underline{\varepsilon} = \left(\varepsilon_{11}\ \varepsilon_{22}\ \varepsilon_{33}\ 2\varepsilon_{12}\ 2\varepsilon_{13}\ 2\varepsilon_{23}\right)^T \qquad (4.61)$$

Both the stress and strain vectors are split into volumetric and deviatoric parts for the stress update using the following transformations

$$\begin{bmatrix} \underline{e} \\ \varepsilon_v \end{bmatrix} = \underline{\underline{C}}_s \underline{\varepsilon} \ , \quad \underline{\underline{C}}_s = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.62}$$

$$\begin{bmatrix} \underline{e} \\ \varepsilon_v \end{bmatrix} = \left(e_{11}\ e_{22}\ e_{33}\ e_{12}\ e_{13}\ e_{23}\ \varepsilon_v\right)^T \tag{4.63}$$

and

$$\begin{bmatrix} \underline{S} \\ \sigma_m \end{bmatrix} = \underline{\underline{C}}_{sa} \underline{\sigma}, \quad \underline{\underline{C}}_{sa} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \end{bmatrix} \tag{4.64}$$

$$\begin{bmatrix} \underline{S} \\ \sigma_m \end{bmatrix} = \left(S_{11}\ S_{22}\ S_{33}\ S_{12}\ S_{13}\ S_{23}\ \sigma_m\right)^T \tag{4.65}$$

The deviatoric and volumetric parts are therefore stored in the same vector for convenience.

A similar transformation is used to change from a deviatoric and volumetric stress state back to $\underline{\sigma}$ and is

$$\underline{\sigma} = \underline{\underline{C}}\begin{bmatrix}\underline{S}\\ \sigma_m\end{bmatrix}, \quad \underline{\underline{C}} = \begin{bmatrix}1&0&0&0&0&0&1\\0&1&0&0&0&0&1\\0&0&1&0&0&0&1\\0&0&0&1&0&0&0\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\end{bmatrix} \qquad (4.66)$$

The above transformations can also be used for the transformation of the elastic and consistent moduli. The elastic stress vector can be calculated as

$$\begin{bmatrix}\underline{S}\\ \sigma_m\end{bmatrix} = \left[\begin{array}{c|c}\underline{\underline{G}} & 0\\ \hline 0 & K(\rho_{n-1})\end{array}\right]\begin{bmatrix}\underline{e}^e\\ \varepsilon_v{}^e\end{bmatrix}$$

$$\underline{\underline{G}} = \begin{bmatrix}2G(\rho_{n-1})&0&0&0&0&0\\0&2G(\rho_{n-1})&0&0&0&0\\0&0&2G(\rho_{n-1})&0&0&0\\0&0&0&2G(\rho_{n-1})&0&0\\0&0&0&0&2G(\rho_{n-1})&0\\0&0&0&0&0&2G(\rho_{n-1})\end{bmatrix} \qquad ...(4.67)$$

where $\underline{\underline{G}}$ is the deviatoric elastic matrix. The elastic matrix, $\underline{\underline{D}}_{el}$, can be calculated from (4.67) using the transformation

$$\underline{\underline{D}}_{el} = \underline{\underline{C}}\left[\begin{array}{c|c}\underline{\underline{G}} & 0\\ \hline 0 & K(\rho_{n-1})\end{array}\right]\underline{\underline{C}}_s \qquad (4.68)$$

The same procedure can be done for the deviatoric consistent moduli. Equation (4.43) can be written in vector notation as

$$\begin{bmatrix}d\underline{S}\\ d\sigma_m\end{bmatrix} = \left[\begin{array}{c|c}\underline{\underline{A}} & \underline{B}\\ \hline \underline{C}^T & D\end{array}\right]\begin{bmatrix}d\underline{e}\\ d\varepsilon_v\end{bmatrix} \qquad (4.69)$$

and the consistent modulus, $\underline{\underline{D}}_{tan}$, is then obtained from

$$\underline{\underline{D}}_{tan} = \underline{\underline{C}}\left[\begin{array}{c|c} \underline{\underline{A}} & \underline{B} \\ \hline \underline{C}^T & D \end{array}\right]\underline{\underline{C}}_s \qquad (4.70)$$

4.4 CALIBRATION OF MATERIAL MODEL FROM ALUMINIUM TEST DATA

The material model contains a number of density dependent variables that have to be defined for a particular material. These density dependent variables include $K(\rho)$ and $G(\rho)$, the bulk and shear moduli, respectively, and the variables that define the shape of the yield surfaces in terms of relative density, $\alpha(\rho)$, $k(\rho)$ and $C_i(\rho)$. The model was calibrated from the data reported by Watson(1993).

Watson(1993:2074) defined equations that fitted the experimental data for the dependence of the bulk modulus and shear modulus on relative density for aluminium powder closely. These equations were defined in terms of Lame's constants from standard relations as

$$K(\rho) = \lambda(\rho) + \frac{2}{3}\mu(\rho), \quad G(\rho) = \mu(\rho) \qquad (4.71)$$

where

$$\mu(\rho) = \mu_1 e^{\mu_2 \rho}, \quad \lambda(\rho) = \lambda_1 e^{\lambda_2 \rho} \qquad (4.72)$$

where μ_1, μ_2, λ_1 and λ_2 are material constants and are listed for aluminium powder in Table 5.1.

Little data could be found for the dependence of the bulk modulus and shear modulus on relative density for other metal powders and therefore some other method of determining the dependence of the elastic moduli on relative density was needed if the material model was to be calibrated for other metal powders. The equations derived by Ramakrishnan(1990:3930) and discussed in Chapter Two were also implemented. They define the elastic moduli as a function of relative density by using the elastic moduli and Poisson's ratio of the fully dense material and are

$$K(\rho) = K\frac{\rho^2}{(1 + b_K(1-\rho))} \qquad (4.73)$$

where $$b_K = \frac{1+\nu}{2(1-2\nu)}$$

$$\text{and} \quad G(\rho) = G\frac{\rho^2}{(1+b_G(1-\rho))} \qquad (4.74)$$

$$\text{where} \quad b_G = \frac{11-19\nu}{4(1+\nu)}$$

where ρ is the relative density and ν, K and G are the Poisson's ratio, bulk modulus and shear modulus of the fully dense material.

Watson(1993:2080) also supplied data that defined the shape of the yield surfaces in terms of density. This data is shown in Appendix D. The variables that define the shape of the yield surfaces, $\alpha(\rho)$, $k(\rho)$ and $C_i(\rho)$ were fitted to this data analytically. Graphs of the data and the curves fitted to this data for $\alpha(\rho)$, $k(\rho)$ and $C_i(\rho)$ are shown in Appendix D. The equations for $\alpha(\rho)$, $k(\rho)$ and $C_i(\rho)$ are listed below as

$$\alpha(\rho) = a + b\rho + c\rho^2 \qquad (4.75)$$

$$k(\rho) = d\rho^f \qquad (4.76)$$

$$C_1(\rho) = g\rho^h \qquad (4.77)$$

$$C_2(\rho) = je^{k\rho} \qquad (4.78)$$

where a, b, c, d, f, g, j and k are material parameters. $C_3(\rho)$ is defined in Equation (3.3).

The model is now calibrated for aluminium powder within the accuracy determined by the fit to the aluminium test data.

CHAPTER 5

MATERIAL MODEL VERIFICATION

The numerical implementation was verified using a series of tests that were performed on a eight-noded three-dimensional finite element. The element was constrained as shown in Figure 5.1 with the appropriate boundary conditions (displacements or stresses) applied to the unconstrained element faces to perform the tests required. The ABAQUS input file used for these tests is shown in Appendix G.

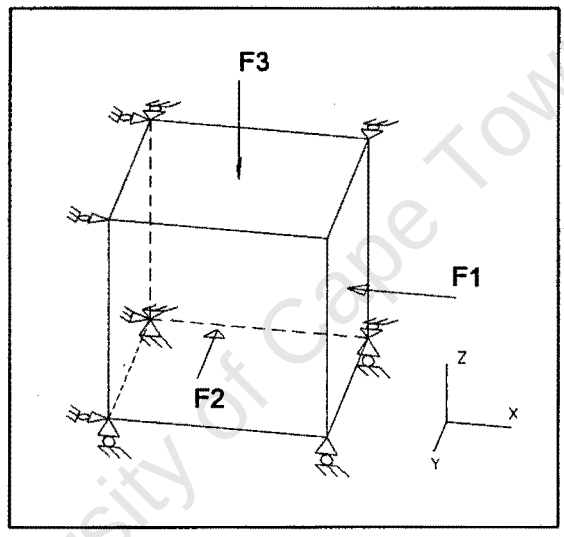

Figure 5.1: Finite element model used for verification tests.

Hydrostatic and constrained compression tests were done to compare the model results with the aluminium test data obtained by Watson(1993). The error involved with the model's stress return algorithm was also checked by changing the number of strain increments to obtain the same stress state. Finally the effect the consistent modulus had on the rate of convergence of the solution was also checked.

5.1 COMPARISON OF MATERIAL MODEL RESULTS WITH ALUMINIUM TEST DATA

The data used for the material parameters in Equations (3.6), (4.72), (4.75)-(4.78) for aluminium powder is shown in Table 5.1 below along with other material data used.

Table 5.1

TABLE OF ALUMINIUM POWDER MATERIAL PARAMETERS FOR MATERIAL MODEL

μ_1	162E6 Pa
μ_2	5.04
λ_1	12E6 Pa
λ_2	8.15
a	0.415
b	0.922
c	-1.115
d	100E6 Pa
f	7
g	-878E6 Pa
h	9
j	55.78E3 Pa
k	7.78
ρ_0	0.52

For the hydrostatic test, faces F1, F2 and F3 in Figure 5.1 were given equal displacements until full density was achieved so that the model could be checked across the full range of relative densities. A comparison of Watson's experimental results and the finite element results for hydrostatic stress versus relative density are shown in Figure 5.2. The finite element results show a good comparison with experiment. Differences in the results may occur due to inaccuracies when curves were fitted to experimental data for the density dependent variables. This was especially the case for the variable $C_2(\rho)$ [See Appendix D], which affects the results for hydrostatic compression.

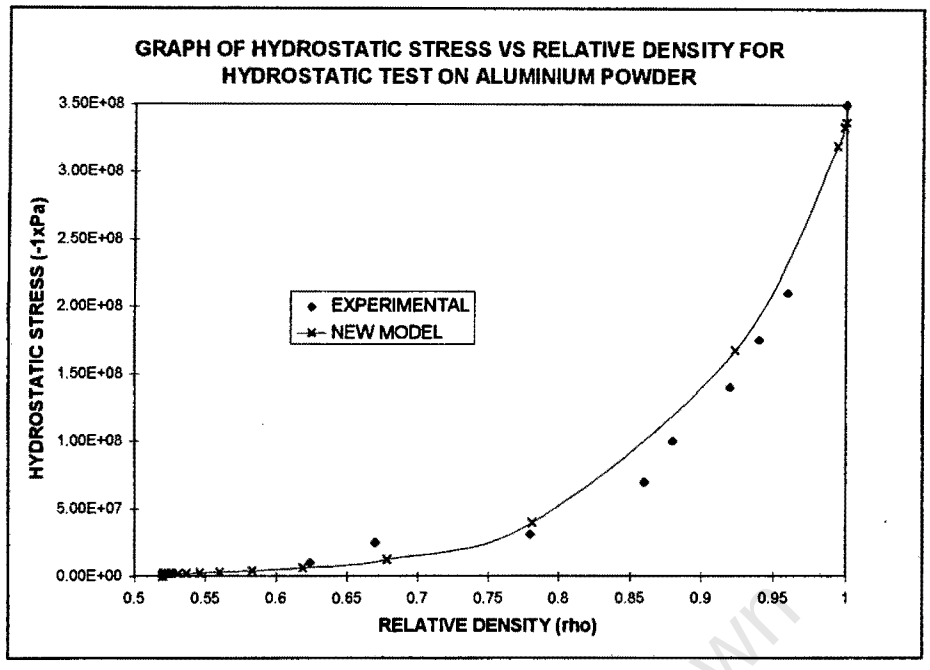

Figure 5.2: Aluminium powder hydrostatic test comparison.

The constrained compression test was performed by constraining faces F1 and F2 and prescribing a displacement on F3. A comparison of Watson's experimental results and the finite element results for axial stress versus relative density are shown in Figure 5.3. The finite element results show good comparison with experiment over most of the relative density range. The largest difference occurs above a relative density of 0.9. This could indicate that the model is not suitable as the relative density approaches full density or it could be due to simplifications made to the original model proposal made by Watson(1993). The test was also done using one thousand fixed increments. There is little difference between the results which indicates that the assumption of keeping the elastic moduli constant for an increment does on affect the results.

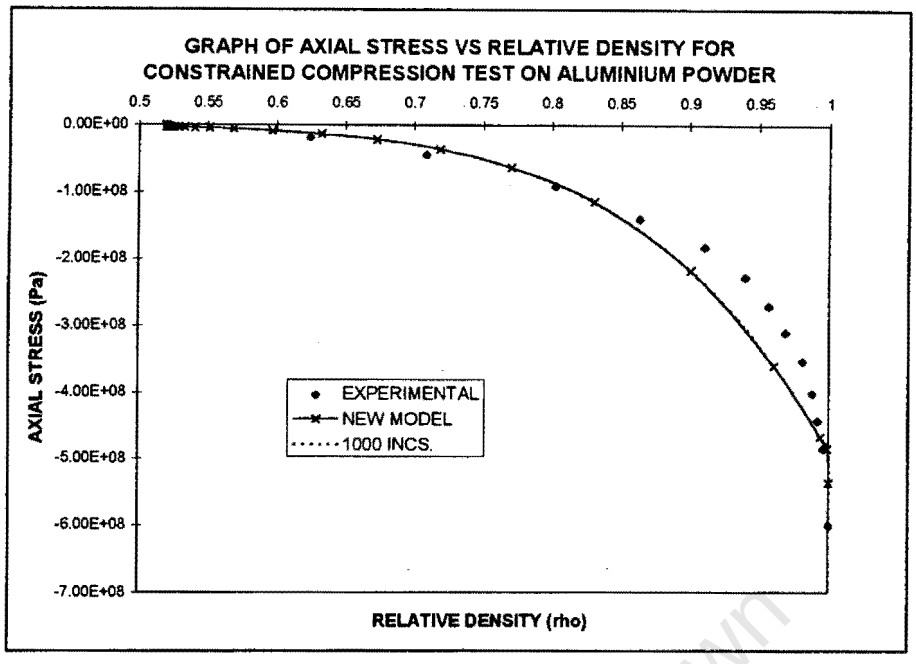

Figure 5.3: Aluminium powder constrained compression test comparison.

Both the hydrostatic and constrained compression finite element results show good agreement with experiment and indicate that the model is behaving correctly.

5.2 STRESS RETURN ALGORITHM ERROR ANALYSIS

A number of tests were performed to check the yield surface position and to quantify the error associated with the stress return onto the yield surface for the shear and consolidation yield surfaces. By quantifying the error produced in varying the number of strain increments used to reach a final solution, an indication of the error associated with the stress return algorithm can be estimated. An arbitrary initial relative density, ρ_0, of 0.7 was chosen for the tests. One test was done on the shear yield surface and four tests were done on the consolidation yield surface.

The results of the test on the shear yield surface are shown in Figure 5.4. An arbitrary strain increment that produced a stress state outside the shear yield surface was used and this was first returned directly to the yield surface. This increment was then divided into four equal size strain increments and returned to the yield surface after each increment. Figure 5.4 shows that in both cases the stress is returned to the yield surface and the final stress state is the same. The return path is also parallel to the S axis as expected and because of this no relative density change occurs, leaving the yield surface position unchanged.

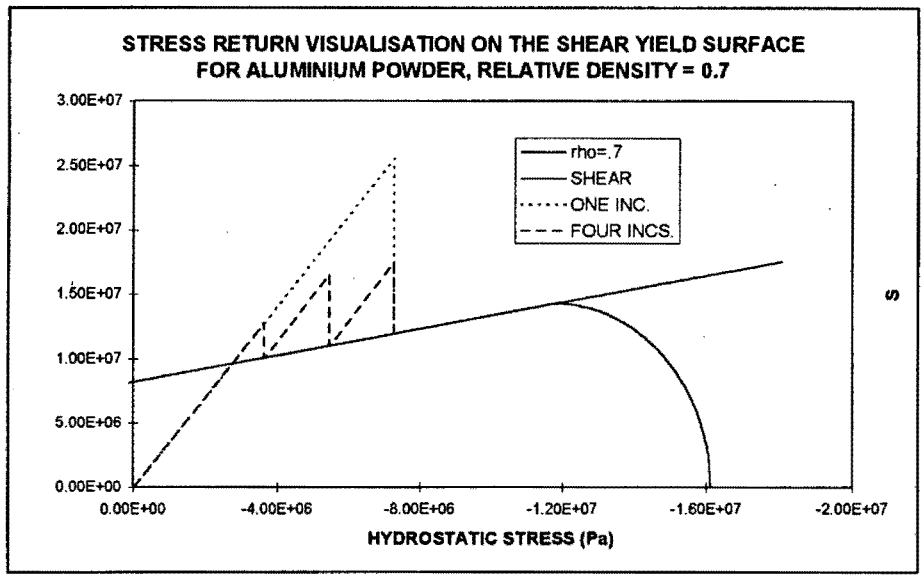

Figure 5.4: Shear yield surface stress return visualisation.

Four different strain increments which produced elastic predictors to cause yielding in different positions over the consolidation yield surface were chosen. These strain increments were applied and returned directly to the yield surface. This was then repeated using four equal increments and 1000 equal increments. The results of the tests on the consolidation yield surface, excluding the hydrostatic test where return is back along the hydrostatic stress axis, are shown in Figure 5.5-Figure 5.7 and indicate that the return path is normal to the yield surface and that the stress point is returned onto the yield surface in all cases. Yielding on the consolidation yield surface produced an increase in relative density in each case and the updated yield surface is indicated in the figures. What can also be seen is that the final stress state differs depending on the number of strain increments used.

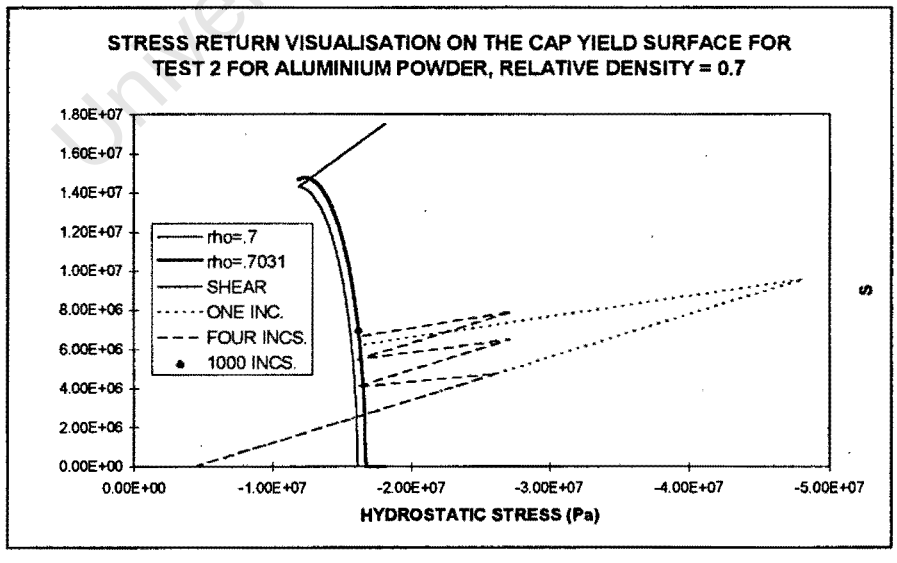

Figure 5.5: Consolidation yield surface stress return visualisation test 2.

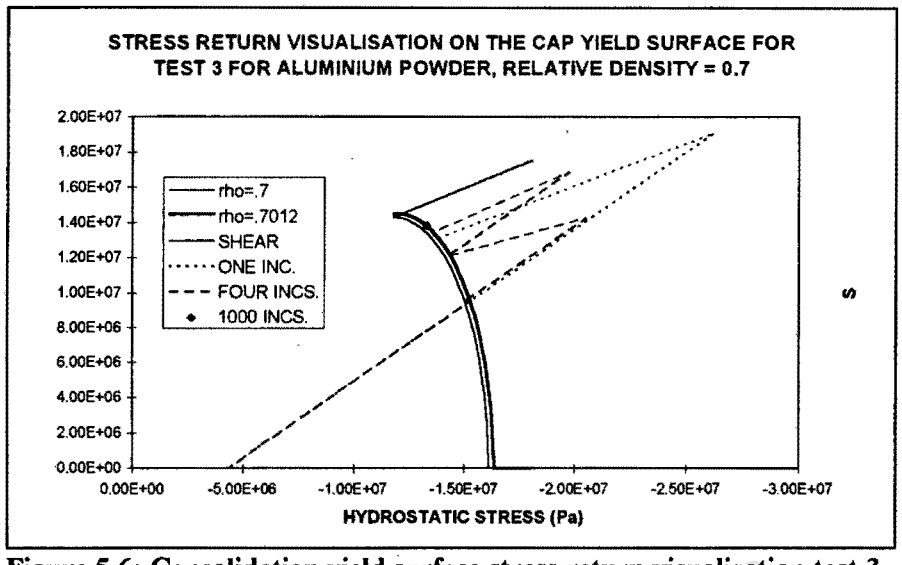

Figure 5.6: Consolidation yield surface stress return visualisation test 3.

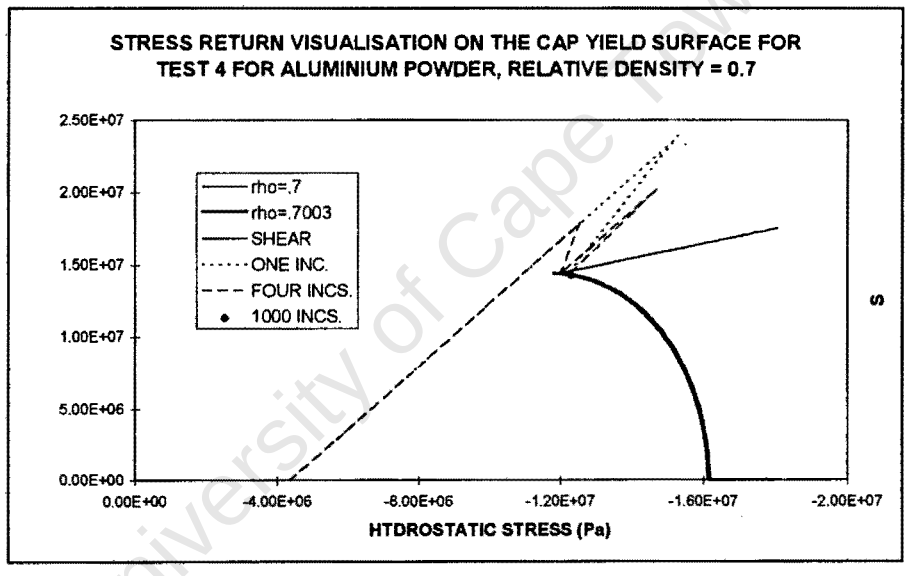

Figure 5.7: Consolidation yield surface stress return visualisation test 4.

The error in the stress return algorithm can be quantified by comparing the results for a different number of strain increments. The final stress state achieved when using 1000 increments during the stress return tests was taken as the exact final stress state. From this the error in the S and σ_m values for the one and four increment tests could be determined. The results are shown in Table 5.2 where Test One is the hydrostatic test and the subsequent test numbers are indicated in Figure 5.5-Figure 5.7.

Table 5.2

TABLE OF % ERROR FOR STRESS RETURN ONTO THE MODEL YIELD SURFACES

	SHEAR		CONSOLIDATION							
			TEST 1		TEST 2		TEST 3		TEST 4	
NO. OF INCS.	σ_m	S	σ_m	S	σ_m	S	σ_m	S	σ_m	S
1	0	0	0	0	.643	10.3	3.96	5.10	.992	.266
4	0	0	0	0	.266	4.05	2.31	2.71	.342	.098
1000	0	0	0	0	0	0	0	0	0	0

The data in Table 5.2 shows that the percentage error associated with the stress return algorithm for the stress states tested are below ten percent. It can also be seen that when the elastic predictor stress path is normal to the yield surface, as in the case of hydrostatic loading, there is zero error associated with the stress return.

5.3 SOLUTION CONVERGENCE RATE

The effect of the consistent modulus on the rate of convergence of the solution was checked by repeating the hydrostatic and constrained compression tests and applying a stress instead of a displacement. The residual forces for each equilibrium iteration done to achieve convergence of the solution for a number of arbitrary increments were plotted and are shown in Figure 5.8 and Figure 5.9 for the hydrostatic test and constrained compression test, respectively. The graphs show that the consistent modulus used produced an acceptable convergence rate.

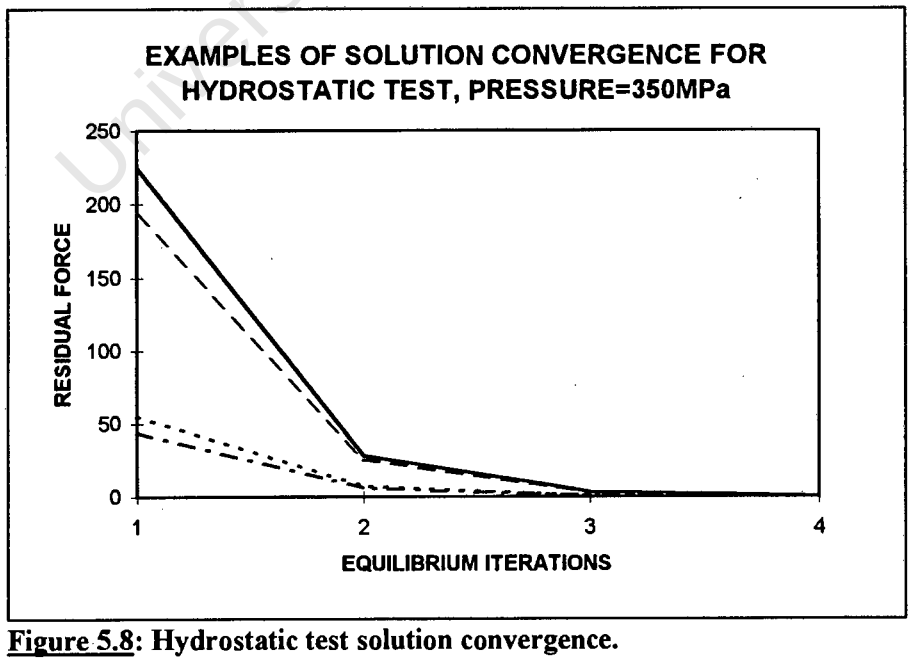

Figure 5.8: Hydrostatic test solution convergence.

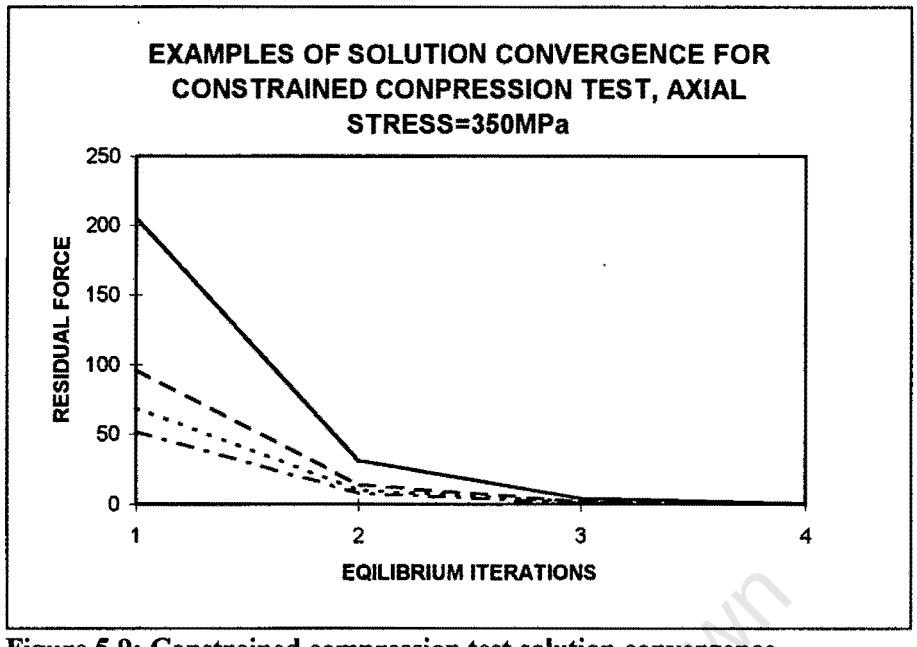

Figure 5.9: Constrained compression test solution convergence.

The procedures used to verify the material model in this chapter have therefore shown that the model can be used with some confidence for modelling metal powder compaction where the behaviour is dominated by consolidation behaviour.

CHAPTER 6

FINITE ELEMENT SIMULATION OF THE COMPACTION OF AN IRON RING

The new constitutive model that was implemented and tested in Chapters Four and Five, was used to simulate the compaction of iron powder ring. Two simulations of the ring compaction were run. The first simulation was with the new constitutive model calibrated for -100 mesh iron powder. The second simulation was with the Porous Metal Plasticity Model which is implemented in ABAQUS. This was done so that the new constitutive model could be compared to a *quadratic* yield surface model. The results of these simulations were compared with experimental results (Kerr, 1994).

6.1 RING COMPACTION PROCESS

The punch and die system used in the compaction of the iron ring is shown schematically in Figure 6.1. The iron powder was compacted from an initial fill height of thirty millimetres down to the final dimensions of the ring using a fixed lower punch and moving upper punch as shown in Figure 6.1. The final dimensions of the compacted preform are shown in Appendix I. The process can be assumed to be essentially quasi-static.

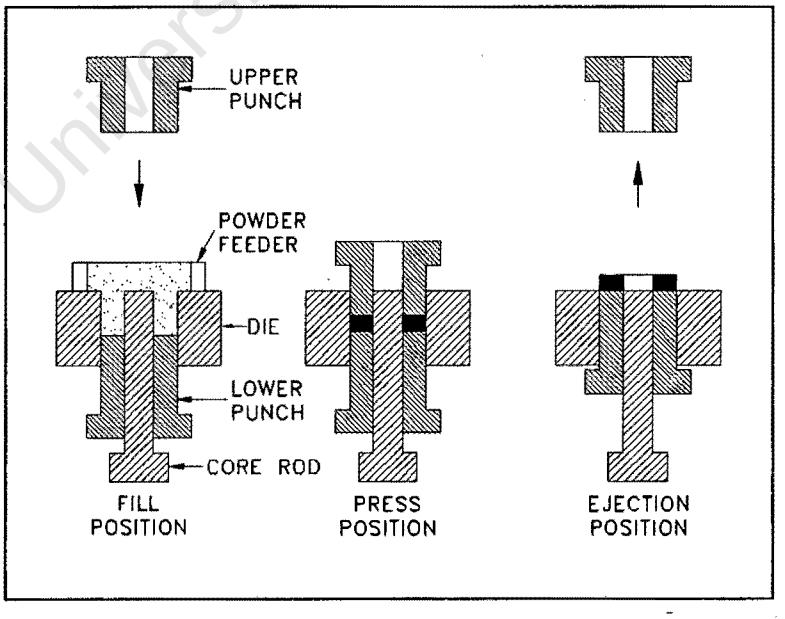

Figure 6.1: Ring compaction process schematic.

The powder used in the compaction process was Hoeganaes ASC100.29 pure iron powder. Zinc stearate was also added to the powder as a lubricant and made up one

percent of the total powder. Powder and compaction data are summarised in Table 6.1 below.

Table 6.1

POWDER DATA AND COMPACTION DATA FROM COMPACTION OF IRON RING (KERR, 1994)

Powder Type	ASC100.29
Percentage Iron	99%
Percentage Lubricant: Zinc Stearate	1%
Initial Density	3.06 g/cm^3
Initial Relative Density, ρ_0	0.4
Estimated Final Density	6.57 g/cm^3
Estimated Final Relative Density	0.82
Compaction Pressure	±300 MPa

6.2 FINITE ELEMENT MODEL OF RING

An axisymmetric finite element model was used to simulate the process described in Section 6.1. The finite element mesh is shown in Figure 6.2 and consisted of axisymmetric linear four-noded elements (CAX4) available in ABAQUS.

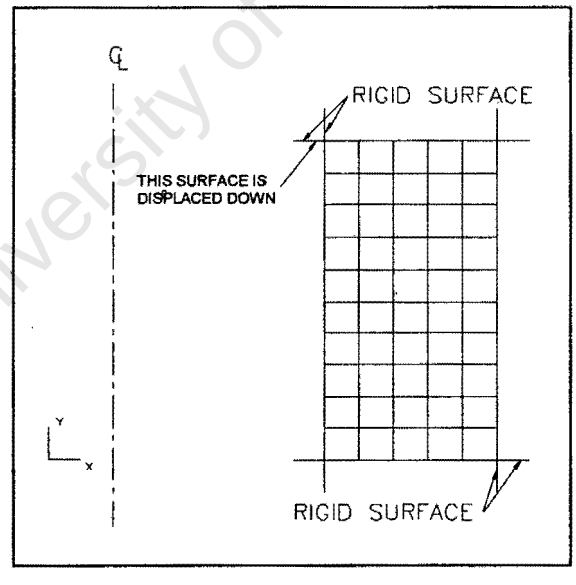

Figure 6.2: Undisplaced axisymmetric mesh for the ring compaction simulation.

The die and punch walls were assumed rigid. The interaction between the punch and die walls and the deforming powder is modelled by special contact elements available in ABAQUS. In this model a coefficient of friction of 0.05 was used. This was estimated from tests done by Tabata(1981:181).

The actual compaction of the powder was modelled by giving the rigid surface representing the upper punch a downward displacement equal to the displacement of the upper punch. This displacement was applied linearly over a certain time period and simulated quasi-static compaction of the powder. The ABAQUS input file used in this simulation is shown in Appendix H.

6.3 CALIBRATION OF MATERIAL MODEL FROM IRON TEST DATA

The Young's modulus, E, and Poisson's ratio, ν, for fully dense iron were obtained from Shigley(1986:510) as

$E = 170$ GPa
$\nu = .3$

The bulk and shear moduli of the fully dense material were determined from E and ν from the standard relations

$$K = \frac{E}{3(1-2\nu)}, \qquad G = \frac{E}{2(1+\nu)} \tag{6.1}$$

The dependence of the bulk and shear moduli on relative density were obtained from Equations (4.73) and (4.74) using K, G and ν.

The variation of $C_1(\rho)$ with density was determined from a data sheet for ASC100.29 powder (Kerr, 1994) which contained data of axial stress versus relative density for constrained compression. This was used to calculate the hydrostatic stress, σ_m, for constrained compression using the theory described in Section 2.2 and used by Watson(1993:2073). $C_1(\rho)$ was then calculated from

$$C_1(\rho) = 3\sigma_m(\rho) \tag{6.2}$$

The other variables, $C_2(\rho)$, $\alpha_N(\rho)$ and $k(\rho)$ were determined from experimental data obtained by Brown(1994) and Schwartz(1969). Schwartz's experimental data included data of hydrostatic stress versus relative density for hydrostatic tests and data for $\alpha_N(\rho)$ and $k(\rho)$. Brown's data was more limited but was obtained for the same Hoeganaes iron powder used in the compaction process. Comparison of the results obtained by Schwartz and Brown for hydrostatic compaction showed that Schwartz's results indicated double the stress for the same relative density. This was probably because the powder used by Schwartz was less compactable. As a result the variables, $C_2(\rho)$, $\alpha_N(\rho)$ and $k(\rho)$ obtained from Schwartz's experiments were divided by two to obtain the final values. This data is shown in Appendix E. The equations of the variables that define the shape of the yield surfaces with relative density, $\alpha(\rho)$, $k(\rho)$ and $C_i(\rho)$ were fitted to this

data. Graphs of the data and the equations for $\alpha(\rho)$, $k(\rho)$ and $C_i(\rho)$ which were fitted to this data are also shown in Appendix E.

The data used for the material parameters in Equations (3.6), (4.73)-(4.78) for iron powder is shown in Table 6.2 below along with other material data used.

Table 6.2

TABLE OF IRON POWDER MATERIAL PARAMETERS FOR THE MATERIAL MODEL

K	141.667 GPa
G	65.3846 GPa
ν	0.3
a	0.9958
b	-0.4191
c	-0.5766
d	250E6 Pa
f	10
g	-3 GPa
h	7.5
j	3.371583E6 Pa
k	6.083256
ρ_0	0.4

6.4 CALIBRATION OF POROUS METAL PLASTICITY MODEL

The Porous Metal Plasticity Model was calibrated by choosing the defining parameters so that the yield surface passed through the constrained compression point on the new constitutive model's yield surface for a particular relative density.

This was done by giving the parameters q_i in Equation (2.8) standard values and then determining σ_y as a function of relative density so that the *quadratic* yield surface passed through the new constitutive model's constrained compression point. Examples of the new model and Porous Metal Plasticity Model yield surfaces at relative densities of 0.6 and 0.9 are shown in Figure 6.3 and Figure 6.4.

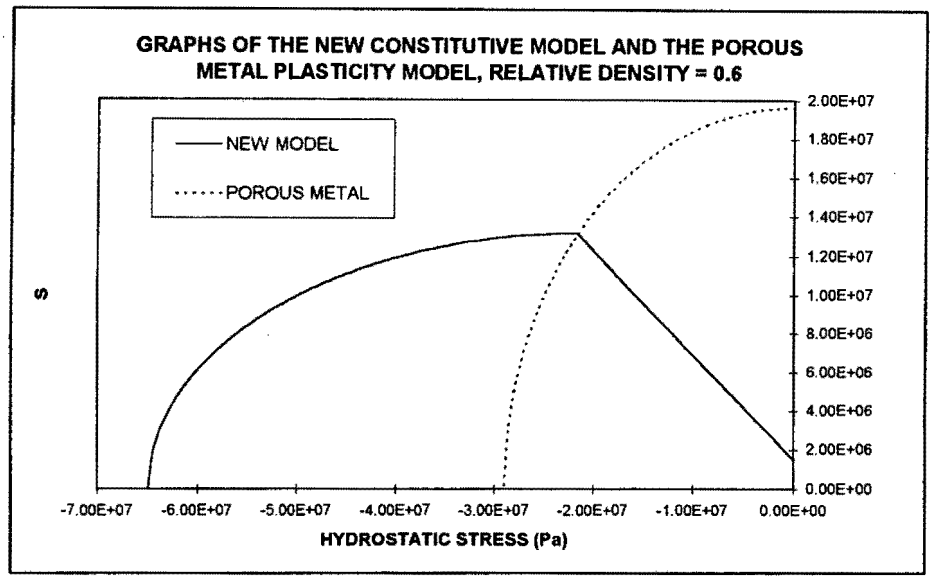

Figure 6.3: Comparison of two yield surfaces at rho=0.6 for iron powder.

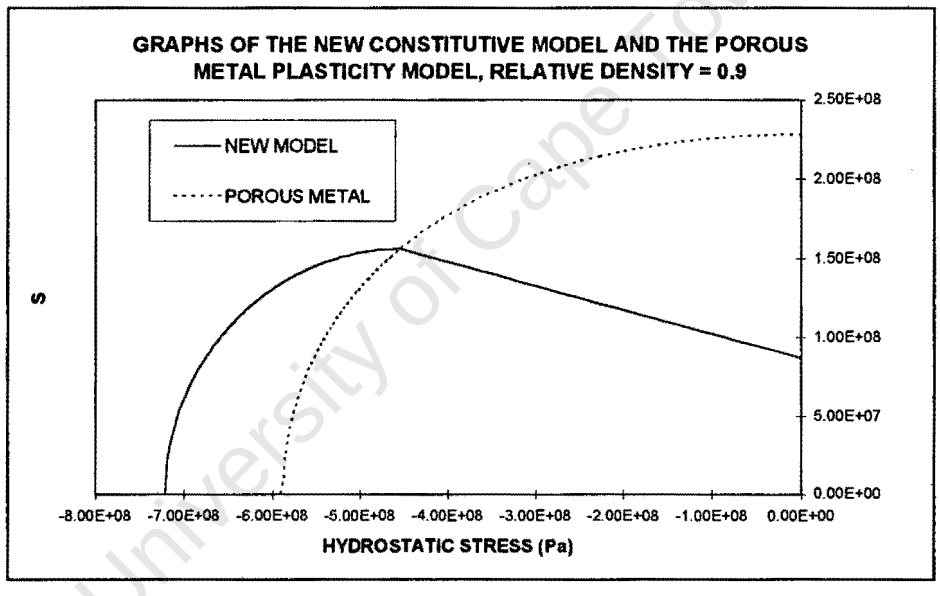

Figure 6.4: Comparison of two yield surfaces at rho=0.9 for iron powder.

The yield stress, σ_y, must be defined in terms of the log plastic strain for the Porous Metal Plasticity Model. Relative density as a function of log plastic strain was obtained from a constrained compression test done on a three-dimensional finite element using the new constitutive model calibrated for iron powder. The parameters used for the Porous Metal Plasticity Model as well as σ_y as a function of log plastic strain are shown in Table 6.3 and Table 6.4 below.

Table 6.3

TABLE OF POROUS METAL PLASTICITY MODEL PARAMETERS

Young's Modulus, E	170 GPa
Poisson's Ratio	0.3
q_1	1.5
q_2	1
q_3	2.25
Initial Relative Density, ρ_0	0.4

Table 6.4

TABLE OF YIELD STRESS, σ_y, VERSUS LOG PLASTIC STRAIN, ε^p_{ln}, FOR POROUS METAL PLASTICITY MODEL

ε^p_{ln}	σ_y
0	2E7
-0.22	4E7
-0.42	8.5E7
-0.55	1.7E8
-0.69	3.05E8
-0.84	4.65E8
-0.94	5.15E8

6.5 DISCUSSION AND ANALYSIS OF RESULTS

Before the simulations of the compaction of the ring were done, hydrostatic and constrained compression tests were carried out on a single three-dimensional finite element as described in Chapter Five for the aluminium powder. The results of these tests were compared to experimental results to give an indication of the performance of the two models before the actual simulations were done.

The results of these tests are plotted in Figure 6.5 and Figure 6.6. Figure 6.5 shows that the new constitutive model agrees extremely well with experimental data for hydrostatic compression while the Porous Metal Plasticity Model does not. The results from the Porous Metal Plasticity Model were expected as its yield surface was fitted to the constrained compression point of the new constitutive model's yield surface. Figure 6.3 and Figure 6.4 show that for hydrostatic loading the Porous Metal Plasticity Model will predict a smaller hydrostatic stress at a particular relative density. The agreement between the two models is better as the relative density approaches one.

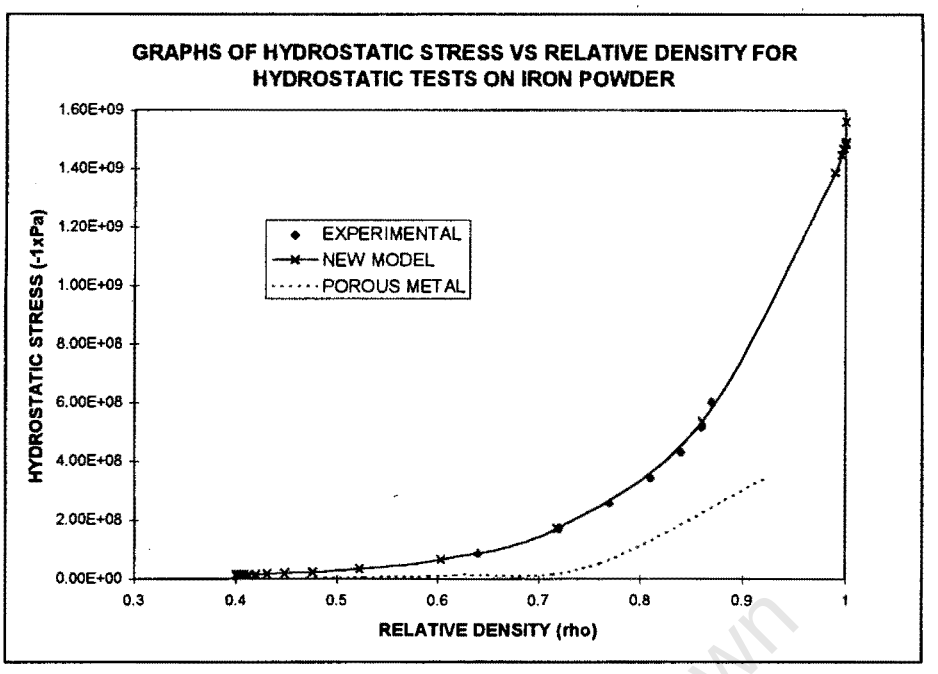

Figure 6.5: Iron powder hydrostatic test comparison.

Figure 6.6 shows that again the new constitutive model agrees better with experiment than the Porous Metal Plasticity Model for constrained compression, although it did predict a higher axial stress for a particular relative density at higher relative densities. This was also noted for the aluminium powder and was attributed to inaccuracies due to simplifications made to the consolidation yield surface at the constrained compression point or the models inability to model compaction at relative densities approaching full density. Again the Porous Metal Plasticity Model predicted a lower stress than that predicted by experiment, however, the agreement with experiment is good for relative densities above 0.9. This is expected as the Porous Metal Plasticity Model was developed for modelling porous metals with relative densities approaching full density. It can be seen from the results of both tests that the new constitutive model agrees well with experiment for most of the relative density range while the Porous Metal Plasticity Model does not. Figures 6.3 and 6.4 indicate that matching the Porous Metal Plasticity Model to the new model under constrained compression conditions may not have given the Porous Metal Plasticity Model the best fit to the iron test data under all conditions and as a result would have affected the results as indicated in Figures 6.5 and 6.6. Matching the Porous Metal Plasticity Model to the iron test data under constrained compression conditions, however, is justified for modelling the iron ring compaction which is under constrained compression loading.

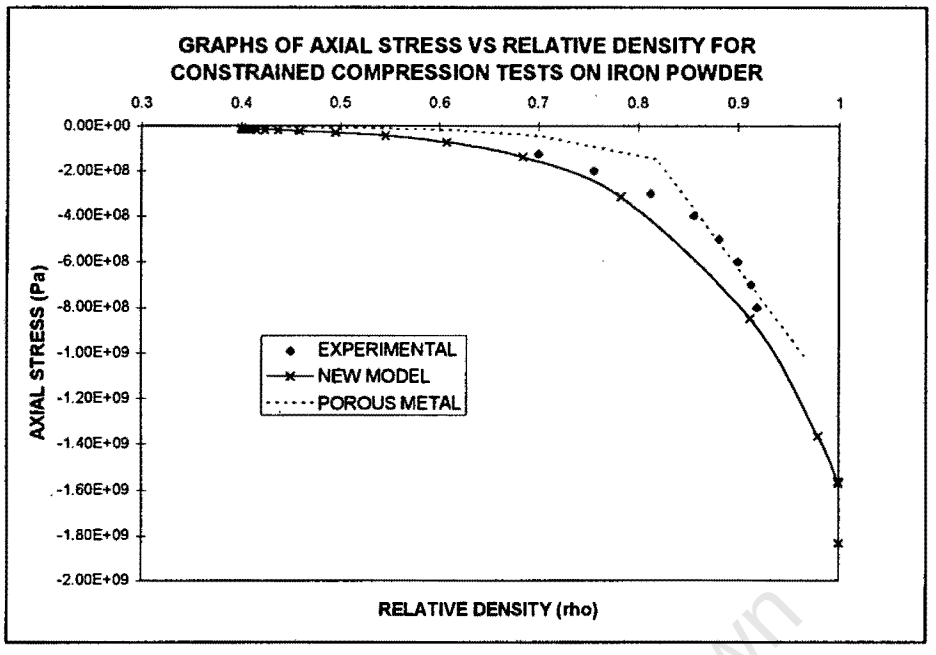

Figure 6.6: Iron powder constrained compression test comparison.

Contour plots of final relative density and axial pressure for the two ring compaction simulations are shown in Figure 6.7-Figure 6.10. Only limited data was available from the actual compaction of ring and was shown in Table 6.1. The final average relative density for the compacted perform was predicted to be approximately 0.82 and the axial stress 300 MPa. From the results shown in the figures below it can be seen that the new constitutive model predicted a final average relative density of 0.813 and an average axial stress of 400 MPa while the Porous Metal Plasticity Model predicted a final average relative density of 0.841 and an average axial stress of 184 MPa.

The relative density prediction from the new constitutive model agrees well with experiment and, as predicted from the constrained compression test results, the axial stress was larger but not significantly. The axial stress predicted by the Porous Metal Plasticity Model was considerably smaller than the experimental result but this was also predicted from the constrained compression results. Comparison of the contour plots for the two simulations shows that the contours for both the simulations were similar for relative density and axial stress values.

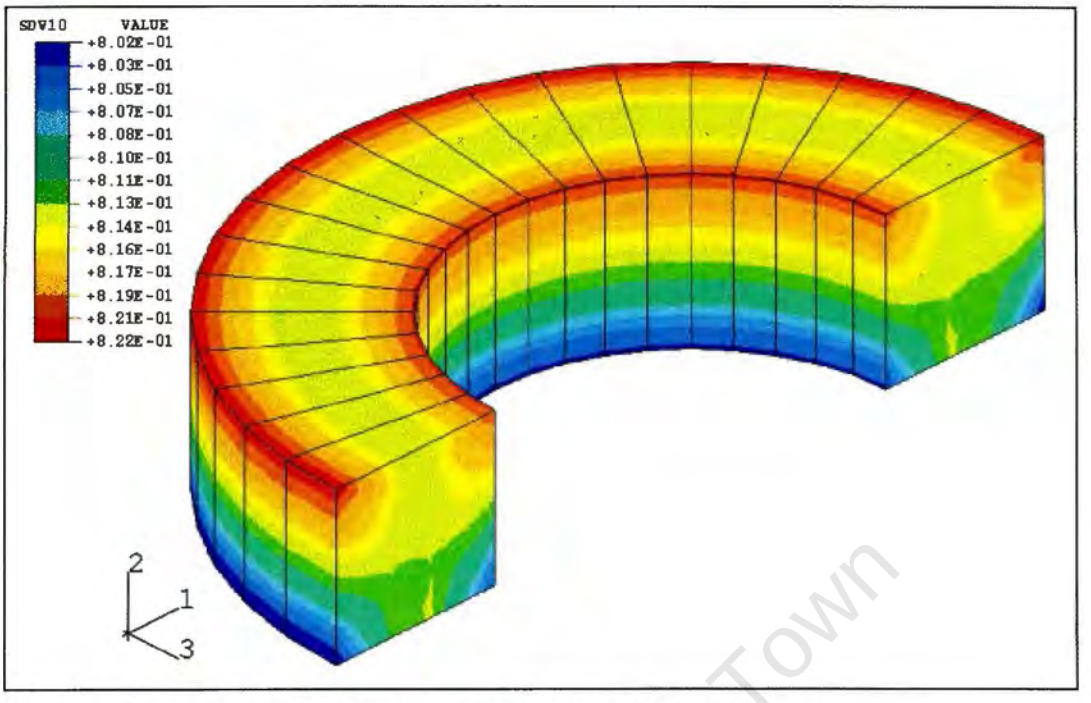

Figure 6.7: Final relative density contour plot for the new model.

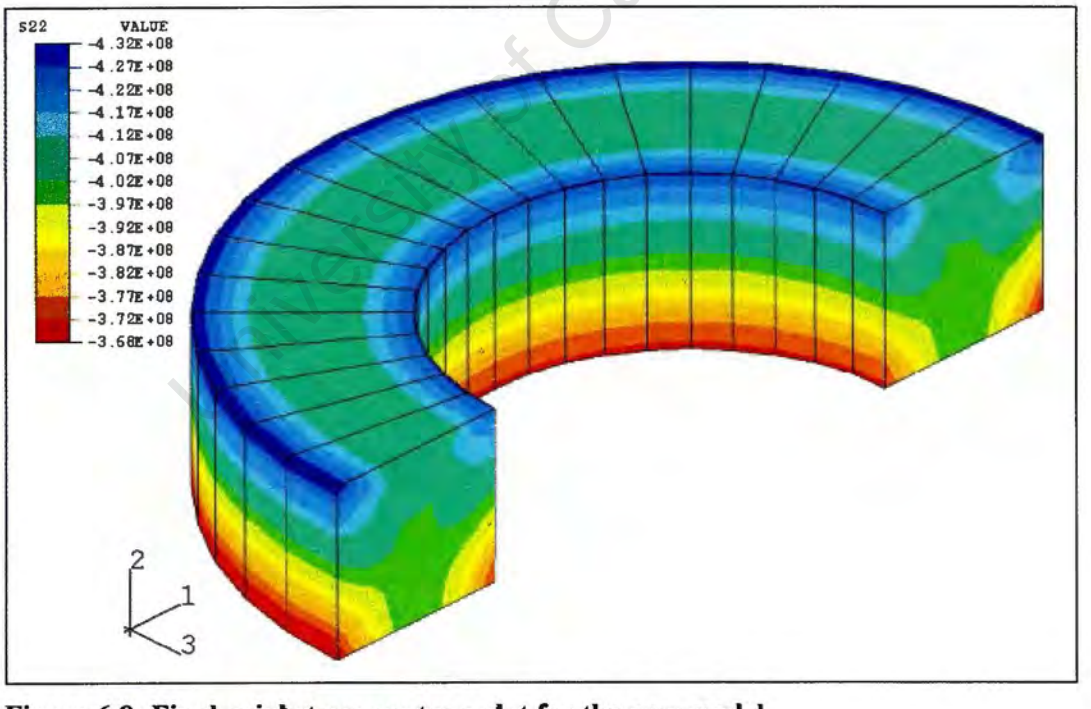

Figure 6.8: Final axial stress contour plot for the new model.

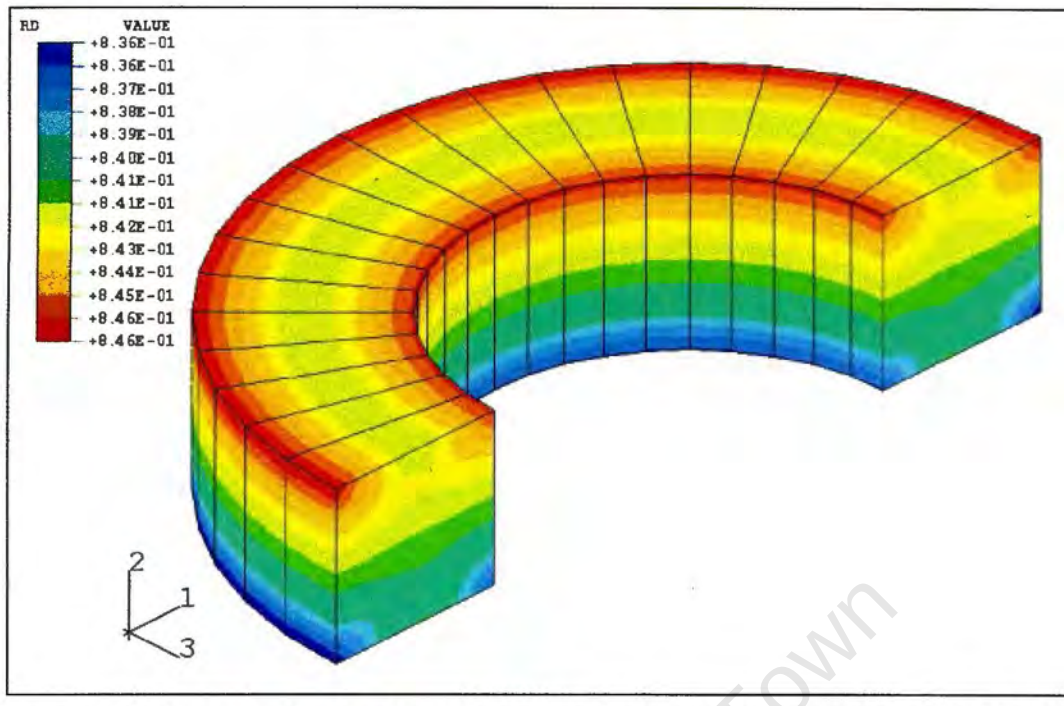

Figure 6.9: Final relative density contour plot for the Porous Metal Plasticity Model.

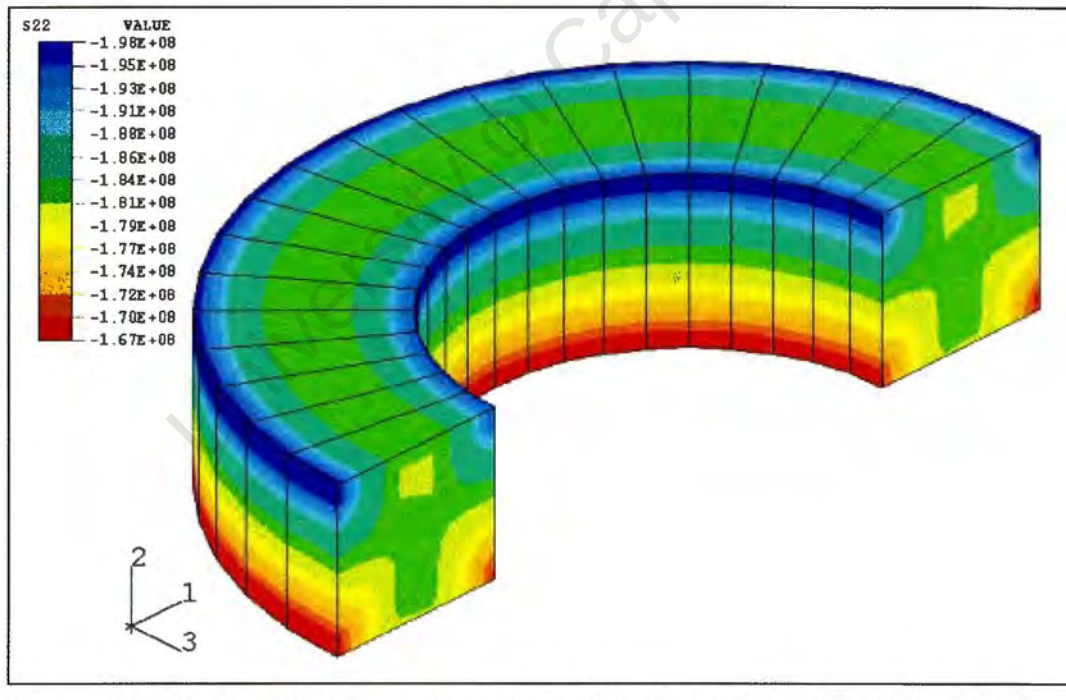

Figure 6.10: Final axial stress contour plot for the Porous Metal Plasticity Model.

CHAPTER 7

CONCLUSIONS

Based on the results of this thesis, the following conclusions can be drawn regarding the performance of the new constitutive model that was implemented in ABAQUS as a FORTRAN 77 User-Material Subroutine for the finite element analysis of metal powder compaction.

The new constitutive model agreed well with experimental results for hydrostatic compaction of both aluminium and iron powder. The new constitutive model also agreed reasonably well with experiment for constrained compression of both aluminium and iron powder especially for lower relative densities up to 0.9. Deviation from experiment above a relative density of 0.9 could be due to simplifications made to the consolidation yield surface at the constrained compression point or the model's inability to model compaction at relative densities approaching full density. Tests would have to be done to determine the actual reason. Similar results were also observed for the ring compaction simulation. The new model was not tested for loading on the shear yield surface but the simplified use of a von Mises flow potential on the shear yield surface was justified for metal powder compaction where compaction occurs mainly on the consolidation yield surface.

A limited comparison between the Porous Metal Plasticity Model and the new constitutive model indicated that the new constitutive model agreed better with experiment than the Porous Metal Plasticity Model, matched to the new model under constrained compression conditions for iron powder, especially for lower relative densities. The Porous Metal Plasticity Model's performance seemed to improve as the relative density approached full density which indicated its suitability for modelling compaction near full density. This result was expected as the Porous Metal Plasticity Model was developed for modelling porous metals with relative densities approaching full density. It was, however, decided that although the choice of matching the Porous Metal Plasticity Model to the new model under constrained compression conditions was justified for the ring compaction simulation, matching the Porous Metal Plasticity Model to the new model under constrained compression conditions may not have given the Porous Metal Plasticity Model the best fit to the iron test data under all conditions and as a result affected the results.

The new constitutive model developed in this thesis can therefore be used with some confidence for the finite element analysis of the compaction of -100 mesh aluminium powder and ASC100.29 iron powder where the powder behaviour is dominated by consolidation behaviour.

Based on the findings and conclusions above, the following recommendations are made for the improvement to the constitutive model developed in this thesis:

1. The reasons for the deviation from experiment at higher relative densities for constrained compression results must be studied in more detail.

2. The new constitutive model must be compared with experimental results of actual relative density profiles obtained using more complicated die geometries.

3. The model must be calibrated for other metal powders with differing morphologies to assess the model's ability to model the compaction of a variety of metal powders.

4. Investigate the effect of different flow potentials on the performance of the shear yield surface and how this affects the coupling between the shear and consolidation yield surfaces.

REFERENCES

Aravas N (1987) 'On the Numerical Integration of a Class of Pressure-Dependent Plasticity Models.' *International Journal for Numerical Methods in Engineering.* vol. 24, p1395-1416

Brookes, JA (1992) 'World Directory and Handbooks of Hardmetals and Hard Materials.' Fifth Edition. International Carbide Data, United Kingdom.

Brown S, Abou-Chedid G (1994) 'Yield Behaviour of Metal Powder Assemblages.' *Journal of the Mechanics and Physics of Solids.* vol. 42, no. 3, p383-399.

Chtourou H, Gakwaya A, Guillot M, Meftah H (1995) 'Implementing a Cap Material Model for the Simulation of Metal Powder Compaction.' *Net Shape Processing of Powder Materials.* ASME International Mechanical Engineering Congress and Exposition, 12-17 November, San Francisco, California, U.S.A.

Dimaggio F, Sandler I (1971) 'Material Model for Granular Soils.' *Journal of Engineering Mechanics Division.* vol. 96, p935-950.

Doraivelu SM, Gegel HL, Gunasekera JS, Malas JC, Morgan JT (1984) 'A New Yield Function For Compressible P/M Materials.' *International Journal of Mechanical Sciences.* vol. 26, no. 9, p527-535.

Fleck NA, Kuhn LT, McMeeking RM (1992) 'Yielding of Metal Powder Bonded by Isolated Contacts.' *The Journal of the Mechanics and Physics of Solids.* vol. 40, no. 5, p1139-1162.

Gollion J, Bouvard D, Stutz P, Grazzini H, Levaillant C, Bauduin P, Cescutti JP (1989) 'On the Rheology of Metal Powder During Cold Compaction.' *Proceedings of the International Conference on Powders and Grains.* 4-8 September, p433-438, Clermond-Ferrand France.

Gurson AL (1977) 'Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I - Yield Criteria and Flow Rules for Porous Ductile Materials.' *J. Eng. Mat. Tech.* vol. 99, p2-15.

Haggblad, H (1993) 'Constitutive Models for Powder Materials'. *Modelling and Simulation of Metal Powder Pressing.* Doctoral Thesis. Lulea University of Technology. p.A1-A10.

HKS Inc. (1995) 'ABAQUS Version 5.5.' Pawtucket RI, USA, Hibbett, Karlsson & Sorenson Inc.

Kamalie H, Tran DV, Gethin DT, Lewis RW (1989) 'Mathematical Modelling Tools for Powder Materials Compaction Processes.' Technical report for G.K.N., Dept. of Civil Engineering, University of Wales, Swansea.

Kerr, J (1994) Sintered Metal Components, Cape Town. Private Communication. 6 October.

Lenel, FV (undated) *Fundamentals and Applications of Powder Metallurgy.* Materials Park, USA, ASM International. p1-13, Lesson 6, Course 15.

Lubliner J (1990) *Plasticity Theory.* Macmillan Publishing Company.

Ortiz M, Simo JC (1986) 'An Analysis of a New Class of Integration Algorithms for Elastoplastic Constitutive Relations.' *International Journal for Numerical Methods in Engineering.* vol. 23, p353-366.

Ramakrishnan N, Arunachalam VS (1990) 'Effective Elastic Moduli of Porous Solids.' *Journal of Material Science.* vol. 25, p3930-3937.

Resende L, Martin JB (1985) 'Formulation of Drucker-Prager Cap Model.' *Journal of Engineering Mechanics.* vol. 8, p839-1570.

Sandler IS, Rubin D (1979) 'An Algorithm and a Modular Subroutine for the Cap Model.' *International Journal for Numerical and Analytical Methods in Geomechanics.* vol. 3, p173-186.

Schwartz EG, Holland AR (1969) 'Determination of Yield Criterion for Iron Powder Undergoing Compaction.' *International Journal of Powder Metallurgy.* vol. 5, no. 1, p.79-86.

Shigley JE (1986) *Mechanical Engineering Design: First Metric Edition.* McGraw-Hill Inc. p510

Tabata T, Masaki S, Kamata K (1981) 'Coefficient of Friction between Metal Powder and Die-Wall During Compaction.' *Powder Metallurgy International.* vol. 13, no. 4, p179-182.

Watson TJ, Wert JA (1993) 'On the Development of Constitutive Relations for Metallic Powders.' *Metallurgical Transactions.* vol. 24A, p2071-2081.

APPENDIX A

DERIVATION OF DENSITY EVOLUTION EQUATION

Equation (3.5) can be written as

$$\frac{d\rho}{\rho} = -d\varepsilon_v^p \qquad \text{(A.1)}$$

where ρ is the relative density and ε^p_v is the volumetric plastic strain and this equation can be integrated as

$$\ln(\rho) + C = -\varepsilon_v^p \qquad \text{(A.2)}$$

where C is a constant. If the conditions $\rho=\rho_0$ for $\varepsilon^p_v=0$ are applied to Equation (A.2), C can be solved for as

$$C = -\ln(\rho_0) \qquad \text{(A.3)}$$

and Equation (A.2) can be rewritten as

$$\ln(\rho) - \ln(\rho_0) = -\varepsilon^p{}_v \qquad \text{(A.4)}$$

which can be simplified as

$$\ln\left(\frac{\rho}{\rho_0}\right) = -\varepsilon_v^p \qquad \text{(A.5)}$$

$$\frac{\rho}{\rho_0} = e^{-\varepsilon_v^p} \qquad \text{(A.6)}$$

$$\rho = \rho_0 e^{-\varepsilon_v^p} \qquad \text{(A.7)}$$

Equations (A.7) and (2.15) are compared in Figure A.1 below using an initial relative of 0.52 used by Watson(1993) for -100 mesh aluminium powder.

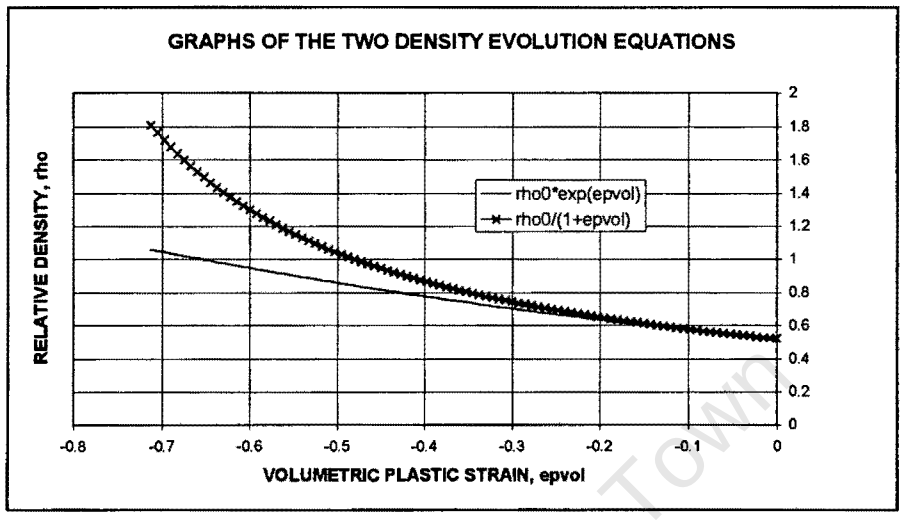

Figure A.1: Comparison of density evolution equations.

APPENDIX B

IMPORTANT CONSOLIDATION YIELD SURFACE DERIVATIVES

$$N(\rho,\sigma_m)=\frac{\partial f_{cons}}{\partial \sigma_m}=6\frac{(C_3(\rho))^2}{(C_2(\rho))^2}(3\sigma_m-C_1(\rho)) \quad \textbf{(B.1)}$$

$$T(\rho)=\frac{\partial^2 f_{cons}}{\partial \sigma_m{}^2}=18\frac{(C_3(\rho))^2}{(C_2(\rho))^2} \quad \textbf{(B.2)}$$

$$P(\rho,\sigma_m)=\frac{\partial f_{cons}}{\partial C_1}=-2\frac{(C_3(\rho))^2}{(C_2(\rho))^2}(3\sigma_m-C_1(\rho)) \quad \textbf{(B.3)}$$

$$Q(\rho,\sigma_m)=\frac{\partial f_{cons}}{\partial C_2}=-2\frac{(C_3(\rho))^2}{(C_2(\rho))^3}(3\sigma_m-C_1(\rho))^2 \quad \textbf{(B.4)}$$

$$R(\rho,\sigma_m)=\frac{\partial f_{cons}}{\partial C_3}=2\frac{(C_3(\rho))}{(C_2(\rho))^2}(3\sigma_m-C_1(\rho))^2-2(C_3(\rho)) \quad \textbf{(B.5)}$$

APPENDIX C

DERIVATION OF EQUATIONS FOR THE PLASTIC MULTIPLIER, λ_α

The equation for the plastic multiplier is derived from

$$\lambda_\alpha^{i+1} = \lambda_\alpha^i - \frac{f\left(\lambda_\alpha^i\right)_\alpha}{f'\left(\lambda_\alpha^i\right)_\alpha} \qquad \text{(C.1)}$$

The yield functions must therefore be determined as a function of lambda.

Shear Yield Surface

The shear yield surface can be written as

$$f\left(\lambda_{shear}\right)_{shear} = S_{n-1} + \Delta S_n + \left(\left(\alpha_N\right)_{n-1} + \left(\Delta\alpha_N\right)_n\right)\left(\left(\sigma_m\right)_{n-1} + \left(\Delta\sigma_m\right)_n\right) - \left(k_{n-1} + \Delta k_n\right) \qquad \text{...(C.2)}$$

For the shear yield surface, $\Delta\alpha_N$ and Δk are always zero as no relative density change occurs when yielding occurs on the surface. Equation (C.2) can therefore be rewritten as

$$f\left(\lambda_{shear}\right)_{shear} = \left(S_{n-1} + \left(\alpha_N\right)_{n-1}\left(\sigma_m\right)_{n-1} - k_{n-1}\right) + \Delta S_n + \left(\alpha_N\right)_{n-1}\left(\Delta\sigma_m\right)_n$$

but $\left(S_{n-1} + \left(\alpha_N\right)_{n-1}\left(\sigma_m\right)_{n-1} - k_{n-1}\right) = \left(f_{shear}\right)_{n-1} = 0$

$$f\left(\lambda_{shear}\right)_{shear} = \Delta S_n + \left(\alpha_N\right)_{n-1}\left(\Delta\sigma_m\right)_n \qquad \text{(C.3)}$$

Substituting for ΔS and $\Delta\sigma_m$ gives

$$f\left(\lambda_{shear}\right)_{shear} = G\left(\rho_{n-1}\right)\Delta e_n - G\left(\rho_{n-1}\right)\lambda_{shear} + \left(\alpha_N\right)_{n-1} K\left(\rho_{n-1}\right)\left(\Delta\varepsilon_v\right)_n \qquad \text{...(C.4)}$$

$$f'(\lambda_{shear})_{shear} = -G(\rho_{n-1}) \tag{C.5}$$

$$\lambda_{shear} = \frac{(f_{shear})^E}{G(\rho_{n-1})} \tag{C.6}$$

Consolidation Yield Surface

The consolidation yield surface can be written as

$$f(\lambda_{cons}) = S_n^{\,2} - \frac{(C_3(\rho_{n-1}) + \Delta C_3(\rho_n))^2}{(C_2(\rho_{n-1}) + \Delta C_2(\rho_n))^2}\left(\begin{matrix}(C_2(\rho_{n-1}) + \Delta C_2(\rho_n))^2 \\ -\left(\begin{matrix}3((\sigma_m)_n) \\ -(C_1(\rho_{n-1}) + \Delta C_1(\rho_n))\end{matrix}\right)^2\end{matrix}\right) \tag{C.7}$$

Substituting for S_n and $(\sigma_m)_n$ from Equations (4.25) and (4.30) and for ΔC_i from

$$\begin{aligned}\Delta C_1 &= \frac{\partial C_1}{\partial \rho}\frac{\partial \rho}{\partial \varepsilon_v^p}\Delta\varepsilon_v^p \\ &= E\Delta\varepsilon_v^p \\ &= E\lambda_{cons}N(\rho,\sigma_m)\end{aligned} \tag{C.8}$$

$$\begin{aligned}\Delta C_2 &= \frac{\partial C_2}{\partial \rho}\frac{\partial \rho}{\partial \varepsilon_v^p}\Delta\varepsilon_v^p \\ &= F\Delta\varepsilon_v^p \\ &= F\lambda_{cons}N(\rho,\sigma_m)\end{aligned} \tag{C.9}$$

$$\begin{aligned}\Delta C_3 &= \frac{\partial C_3}{\partial \rho}\frac{\partial \rho}{\partial \varepsilon_v^p}\Delta\varepsilon_v^p \\ &= M\Delta\varepsilon_v^p \\ &= M\lambda_{cons}N(\rho,\sigma_m)\end{aligned} \tag{C.10}$$

Equation (C.7) can be written as

$$f(\lambda_{cons}) = \left(\frac{S^{<E>}}{1+2\cdot G(\rho_{n-1})\cdot\lambda_{cons}}\right)^2 \ldots + \left[\frac{\left(C_3(\rho_{n-1}) + M(\rho_n)\cdot\lambda_{cons}\cdot N(\rho,\sigma_m)_n\right)^2}{\left(C_2(\rho_{n-1}) + F(\rho_n)\cdot\lambda_{cons}\cdot N(\rho,\sigma_m)_n\right)^2}\cdot\left[\left(C_2(\rho_{n-1}) + F(\rho_n)\cdot\lambda_{cons}\cdot N(\rho,\sigma_m)_n\right)^2 \ldots + \left[3\cdot\left(\sigma_m^{<E>} - K(\rho_{n-1})\cdot\lambda_{cons}\cdot N(\rho,\sigma_m)_n\right) \ldots + \left(C_1(\rho_{n-1}) + E(\rho_n)\cdot\lambda_{cons}\cdot N(\rho,\sigma_m)_n\right)\right]^2\right]\right] \quad \ldots(C.11)$$

where $S^{<E>}$ and $\sigma_m^{<E>}$ are the deviatoric and volumetric elastic predictors respectively. Equation (C.11) can be differentiated with respect to lambda to give

$$\frac{df(\lambda_{cons})}{d\lambda_{cons}} = -4\cdot\frac{{s_n}^2}{\left(1+2\cdot G(\rho_{n-1})\cdot\lambda_{cons}\right)}\cdot G(\rho_{n-1}) \ldots + \left[2\cdot\frac{\left(C_3(\rho_n)\right)}{\left(C_2(\rho_n)\right)^2}\right]\cdot\left[\left(C_2(\rho_n)\right)^2 - \left(3\cdot\sigma_{m_n} - C_1(\rho_n)\right)^2\right]\cdot M(\rho_n)\cdot\left[N(\rho,\sigma_m)_n + \lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right] \ldots + \left[2\cdot\frac{\left(C_3(\rho_n)\right)^2}{\left(C_2(\rho_n)\right)^3}\cdot\left[\left(C_2(\rho_n)\right)^2 - \left(3\cdot\sigma_{m_n} - C_1(\rho_n)\right)^2\right]\cdot F(\rho_n)\cdot\left[N(\rho,\sigma_m)_n + \lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right] \ldots + \left[-\frac{\left(C_3(\rho_n)\right)^2}{\left(C_2(\rho_n)\right)^2}\cdot\left[2\cdot\left(C_2(\rho_n)\right)\cdot F(\rho_n)\cdot\left[N(\rho,\sigma_m)_n + \lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right]\right] \ldots + \frac{\left(C_3(\rho_n)\right)^2}{\left(C_2(\rho_n)\right)^2}\cdot\left[2\cdot\left(3\cdot\sigma_{m_n} - C_1(\rho_n)\right)\cdot\left[-3\cdot K(\rho_{n-1})\cdot\left[N(\rho,\sigma_m)_n + \lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right] \ldots + \left[E(\rho_n)\cdot\left[N(\rho,\sigma_m)_n + \lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right]\right]\right]\right]\right] \quad \ldots(C.12)$$

The term $\frac{dN}{d\lambda}$ is calculated below using Equation (B.1) in the form

$$N(\lambda_{cons})_n = 6\cdot\frac{\left(C_3(\rho_{n-1}) + M(\rho_n)\cdot\lambda_{cons}\cdot N(\lambda_{cons})_n\right)^2}{\left(C_2(\rho_{n-1}) + F(\rho_n)\cdot\lambda_{cons}\cdot N(\lambda_{cons})_n\right)^2}\cdot\left[3\cdot\left(\sigma_m^{<E>} - K(\rho_{n-1})\cdot\lambda_{cons}\cdot N(\lambda_{cons})_n\right) \ldots + \left(-C_1(\rho_{n-1}) - E(\rho_n)\cdot\lambda_{cons}\cdot N(\lambda_{cons})_n\right)\right] \quad (C.13)$$

This equation is then differentiated with respect to lambda to give

$$\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n = 6\cdot\frac{\begin{bmatrix}2\cdot C_3(\rho_n)\cdot(C_2(\rho_n))^2\cdot M(\rho_n)\cdot\left[N(\rho,\sigma_m)_n+\lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right]\ldots\\ +\left[-2\cdot C_2(\rho_n)\cdot(C_3(\rho_n))^2\cdot F(\rho_n)\cdot\left[N(\rho,\sigma_m)_n+\lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right]\right]\end{bmatrix}}{(C_2(\rho_n))^4}\cdot\left(3\cdot\sigma_{m_n}-C_1(\rho_n)\right)\ldots$$
$$+6\cdot\frac{(C_3(\rho_n))^2}{(C_2(\rho_n))^2}\cdot\begin{bmatrix}-3\cdot K(\rho_{n-1})\cdot\left[N(\rho,\sigma_m)_n+\lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right]\ldots\\ +-\left[E(\rho_n)\cdot\left[N(\rho,\sigma_m)_n+\lambda_{cons}\cdot\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n\right]\right]\end{bmatrix}$$

...(C.14)

The $\frac{dN}{d\lambda}$ terms can be collected together to give

$$\left(\frac{dN(\rho,\sigma_m)}{d\lambda_{cons}}\right)_n = \frac{-6\cdot C_3(\rho_n)\cdot N(\rho,\sigma_m)_n\cdot\begin{bmatrix}\left(6\cdot M(\rho)\cdot\sigma_m\cdot C_2(\rho)-2\cdot M(\rho)\cdot C_1(\rho)\cdot C_2(\rho)-6\cdot C_3(\rho)\cdot F(\rho)\cdot\sigma_m\right)\ldots\\ +2\cdot C_3(\rho)\cdot F(\rho)\cdot C_1(\rho)-3\cdot C_3(\rho)\cdot K(\rho_{n-1})\cdot C_2(\rho)-C_3(\rho)\cdot E(\rho)\cdot C_2(\rho)\end{bmatrix}}{\begin{bmatrix}-C_2(\rho)^3+36\cdot C_3(\rho)\cdot M(\rho)\cdot\lambda_{cons}\cdot\sigma_m\cdot C_2(\rho)-12\cdot C_3(\rho)\cdot M(\rho)\cdot\lambda_{cons}\cdot C_1(\rho)\cdot C_2(\rho)\ldots\\ +-36\cdot C_3(\rho)^2\cdot F(\rho)\cdot\lambda_{cons}\cdot\sigma_m+12\cdot C_3(\rho)^2\cdot F(\rho)\cdot\lambda_{cons}\cdot C_1(\rho)\ldots\\ +-18\cdot C_3(\rho)^2\cdot K(\rho_{n-1})\cdot\lambda_{cons}\cdot C_2(\rho)-6\cdot C_3(\rho)^2\cdot E(\rho)\cdot\lambda_{cons}\cdot C_2(\rho)\end{bmatrix}}$$

(C.15)

APPENDIX D

ALUMINIUM POWDER DATA AND CALIBRATION CURVES

The data obtained from Watson's experimental work on -100 mesh aluminium powder for the variables defining the shape of the yield surfaces are shown in Table D.1 below.

Table D.1

TABLE OF DENSITY DEPENDENT VARIABLES FOR ALUMINIUM POWDER YIELD SURFACES

ρ	C_1 (MPa)	C_2 (MPa)	C_3 (MPa)	C_7 (MPa)	C_8 (MPa)	C_9 (MPa)	α_N (MPa)	k (MPa)
0.52	0	0	0	0	0	0	0	0
0.624	-23.47	5.562	8.274	17.52	-24.53	8.123	0.576	5.827
0.709	-61.43	16.51	19.71	38.34	-65.31	19.16	0.552	12.22
0.803	-132.4	33.13	36.58	87.47	-140.7	35.40	0.465	23.48
0.864	-209.6	47.94	51.70	150.4	-222.0	49.94	0.402	34.92
0.911	-292.5	59.74	65.36	247.7	-307.8	63.20	0.340	48.69
0.940	-383.8	53.46	77.75	344.5	-394.2	76.26	0.308	61.32
0.957	-476.6	31.04	90.06	414.1	-479.6	89.62	0.299	71.53
0.969	-565.0	11.63	102.9	482.5	-565.4	102.9	0.293	81.55
0.981	-650.1	29.93	116.1	626.1	-652.3	115.8	0.270	97.44
0.988	-734.3	42.77	129.6	758.7	-738.3	129.0	0.256	112.2
0.992	-819.3	43.31	143.1	854.1	-823.1	142.5	0.252	124.4
0.996	-877.7	131.3	160.4	1128	-907.9	156.1	0.227	147.9
1	-994.3	$-\infty$	169.1	∞	-994.3	169.1	0	276.2

Data in this table obtained from Watson(1993).

The graphs of the above data for α_N, k, C_1, C_2 and C_3 and the curves fitted to this data are shown in Figure D.1-Figure D.5. Equations (4.75)-(4.78) and (3.3) for the fitted curves are repeated below.

$$\alpha(\rho) = a + b\rho + c\rho^2$$

$$k(\rho) = d\rho^f$$

$$C_1(\rho) = g\rho^h$$

$$C_2(\rho) = je^{k\rho}$$

$$C_3(\rho) = -\alpha_N(\rho)\frac{C_1(\rho)}{3} + k(\rho)$$

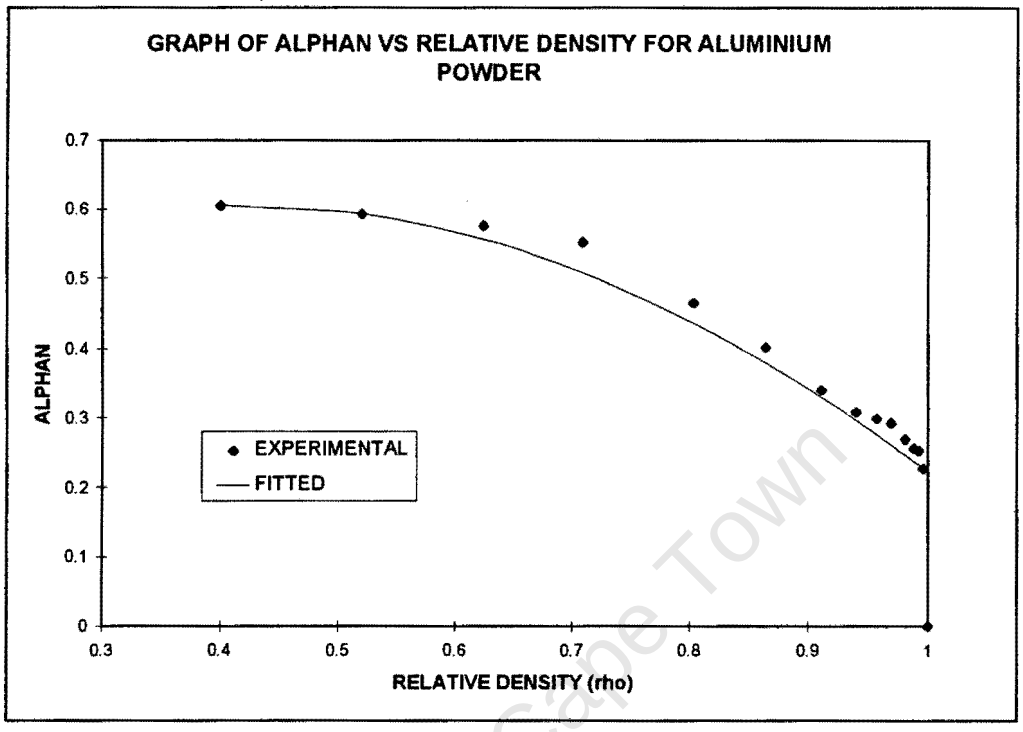

Figure D.1: α_N (Alphan) curve for aluminium powder.

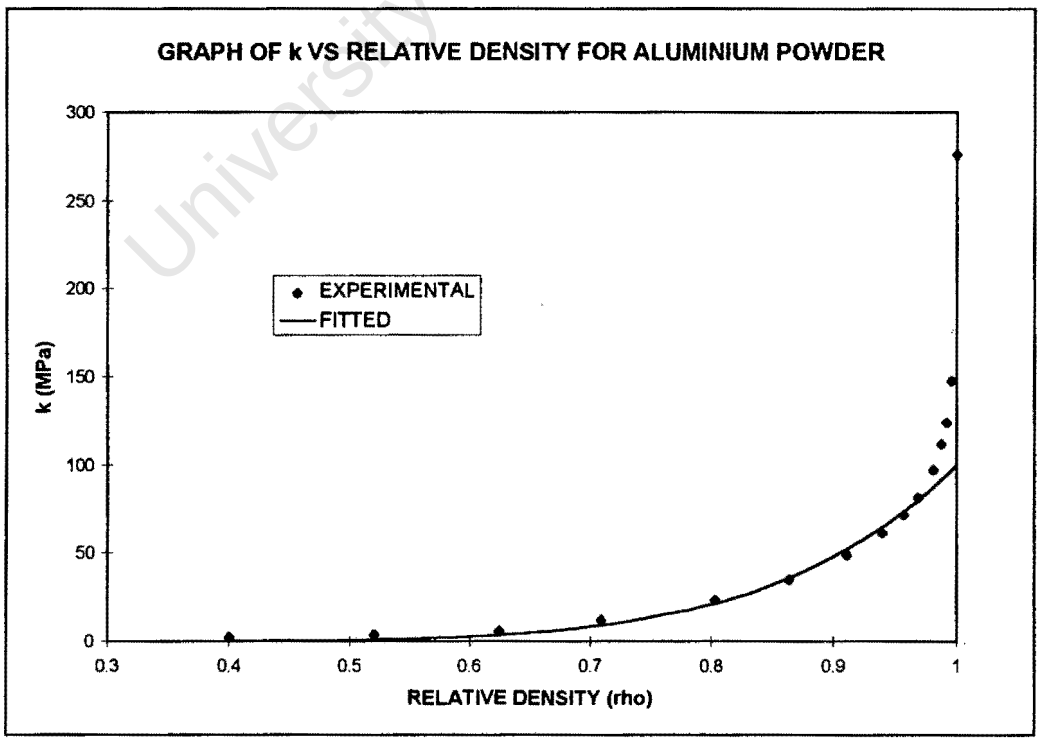

Figure D.2: k curve for aluminium powder.

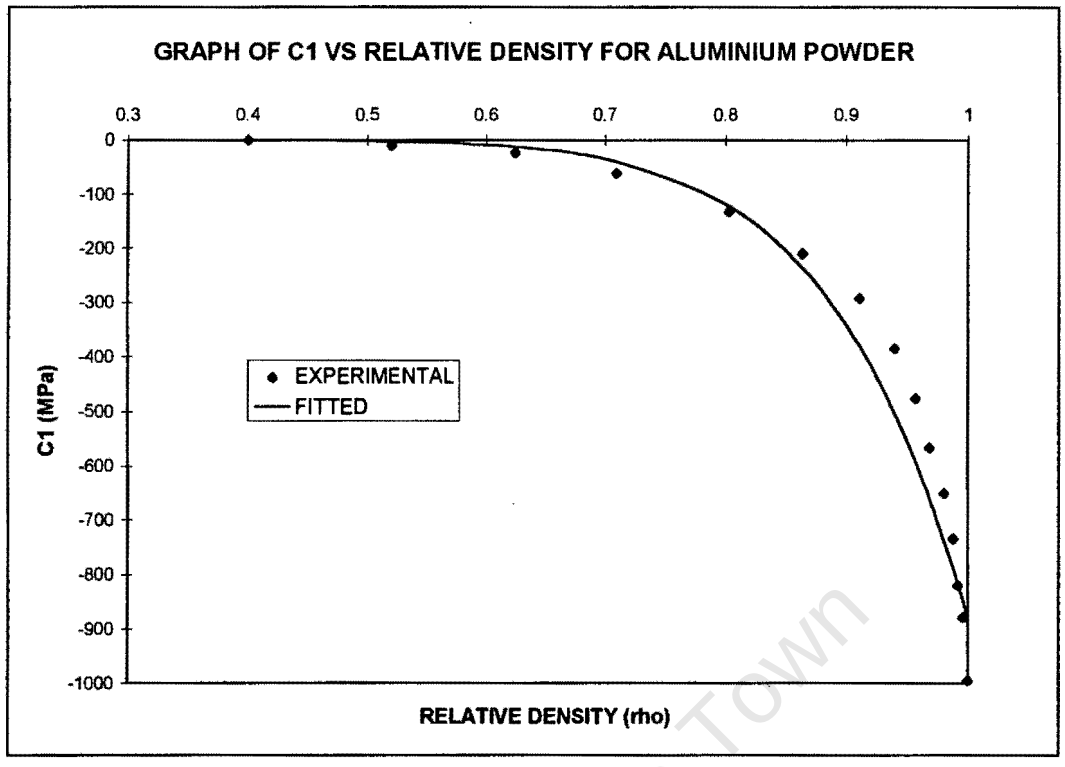

Figure D.3: C_1 curve for aluminium powder.

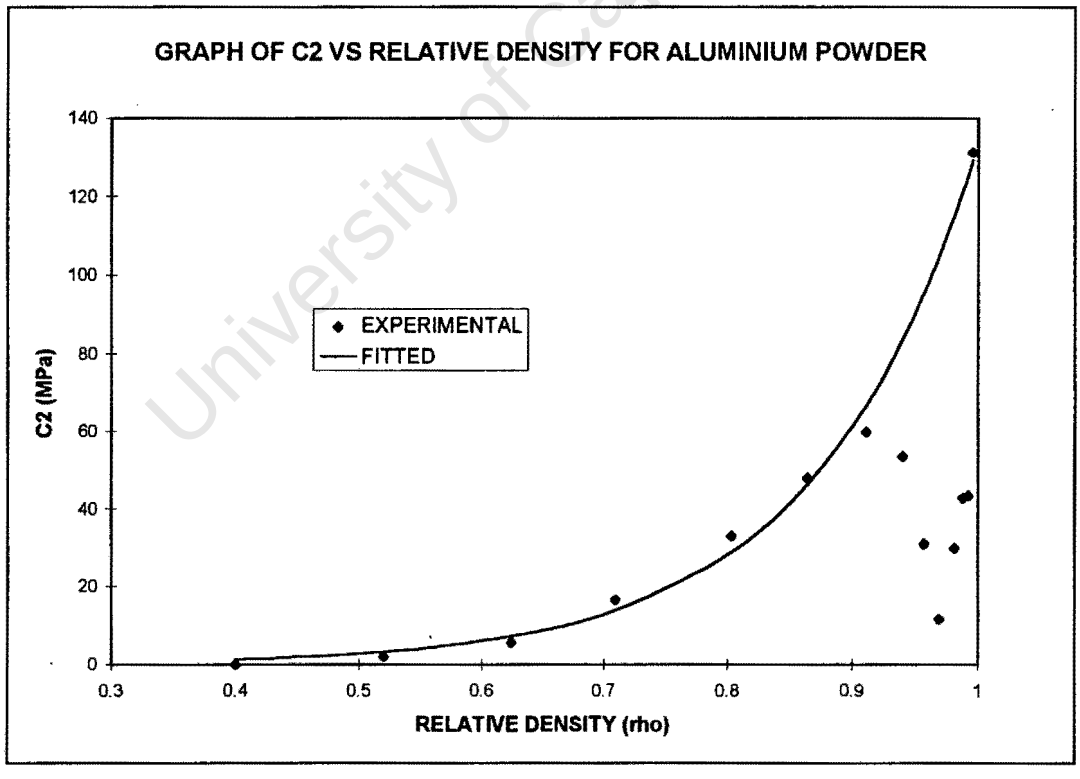

Figure D.4: C_2 curve for aluminium powder.

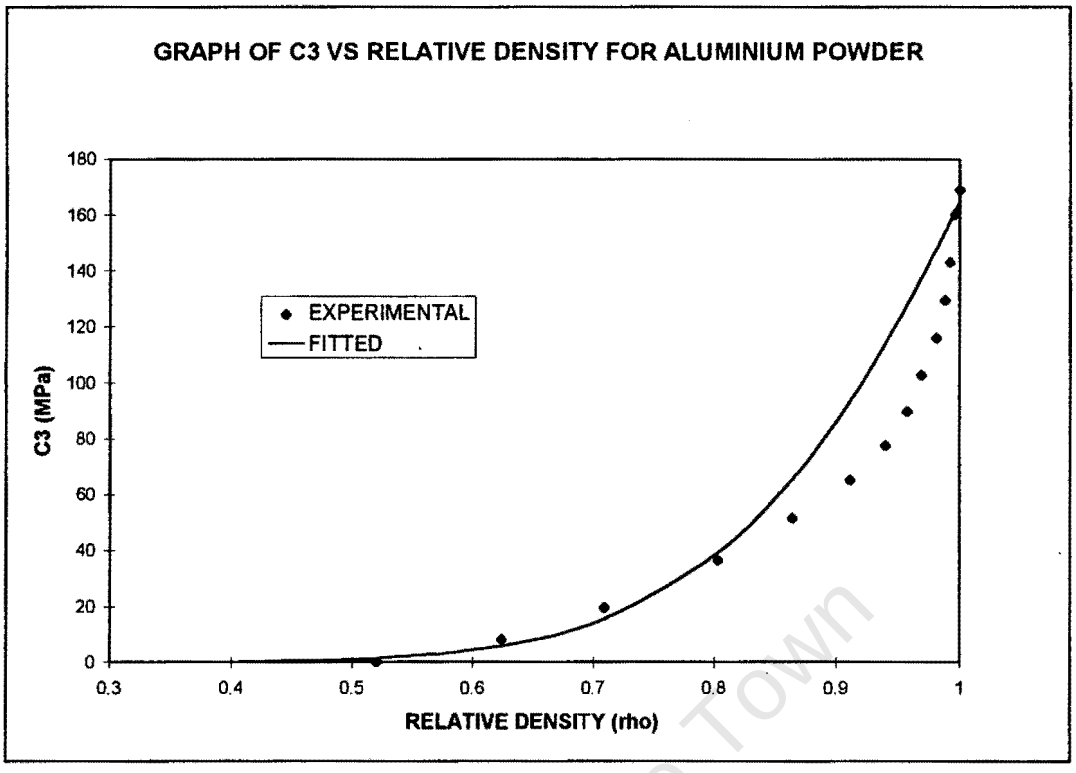

Figure D.5: C_3 curve for aluminium powder.

APPENDIX E

IRON POWDER DATA AND CALIBRATION CURVES

The data obtained from Schwartz's and Brown's experimental work on iron powder for the variables defining the shape of the yield surfaces are shown in Table E.1 below.

Table E.1

TABLE OF DENSITY DEPENDENT VARIABLES FOR IRON POWDER YIELD SURFACES

CONSOLIDATION YIELD SURFACE				SHEAR YIELD SURFACE		
ρ	C_1 (Pa)	C_2 (Pa)	C_3 (Pa)	ρ	α_N (MPa)	k (MPa)
0.7	2.8E8	2.17E8	2.76E7	0.74	0.350	1.60E7
0.76	4.45E8	3.3E8	4.49E7	0.77	0.327	1.92E7
0.813	6.62E8	4.85E8	6.86E7	0.82	0.269	3.47E7
0.856	8.78E8	6.3E8	9.29E7	0.85	0.213	5.26E7
0.881	1.09E9	7.27E8	1.17E8	0.88	0.180	6.53E7
0.9	1.31E9	8.07E8	1.42E8	0.9	0.147	8.98E7
0.912	1.52E9	8.59E8	1.66E8	0.92	0.082	1.32E8
0.919	1.74E9	8.91E8	1..91E8	1	0	

The graphs of the above data for α_N, k, C_1, C_2 and C_3 and the curves fitted to this data are shown in Figure E.1-Figure E.5.

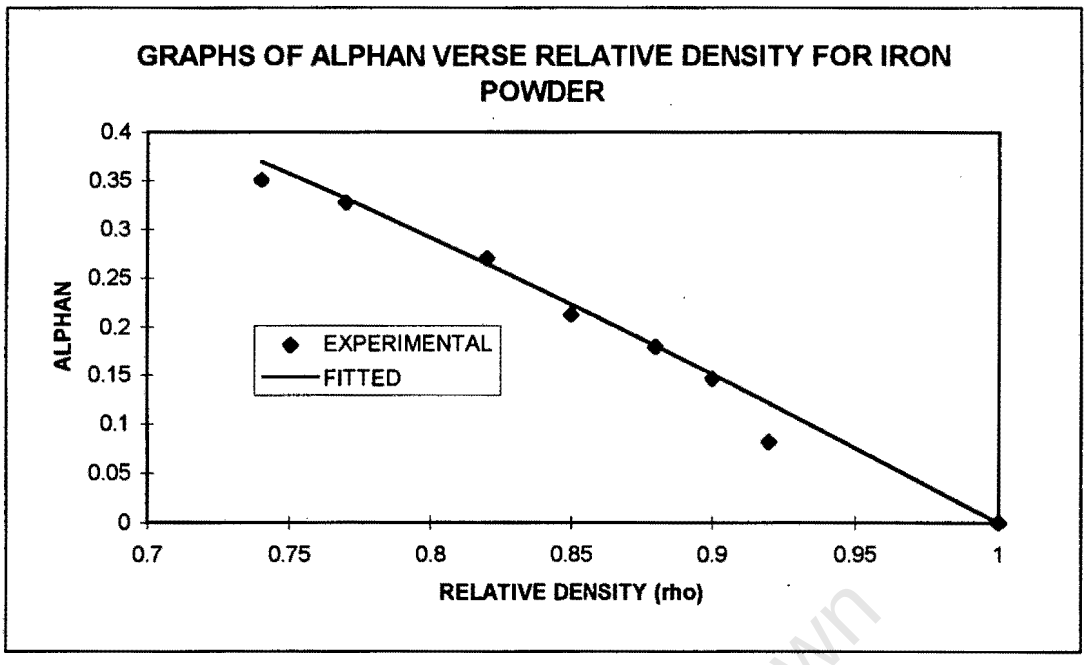

Figure E.1: α_N (Alphan) curve for iron powder.

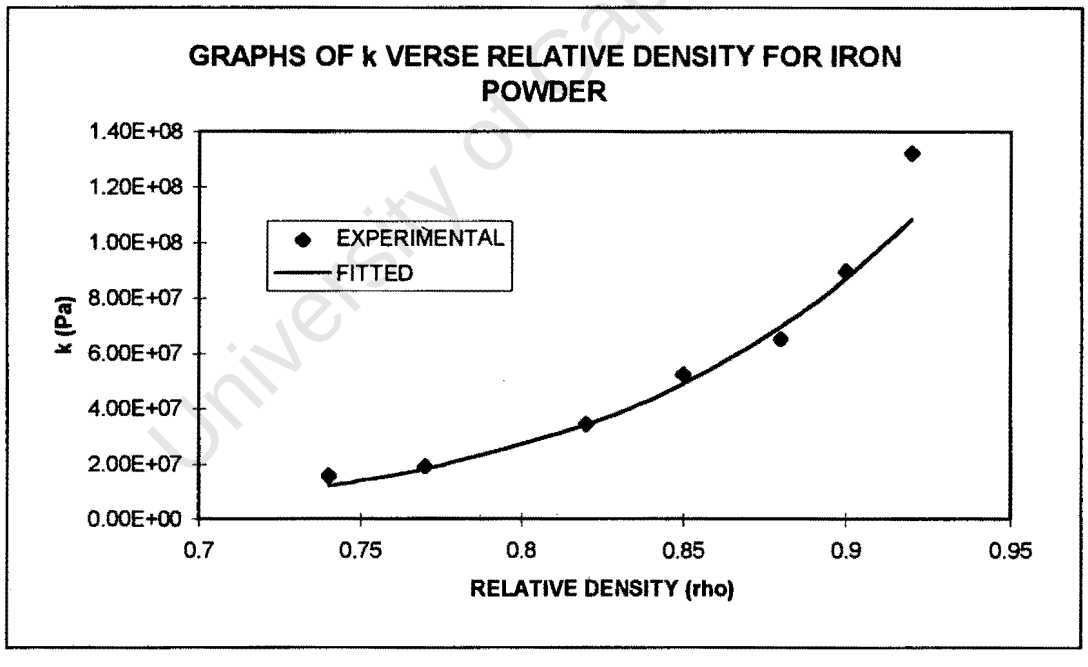

Figure E.2: k curve for iron powder.

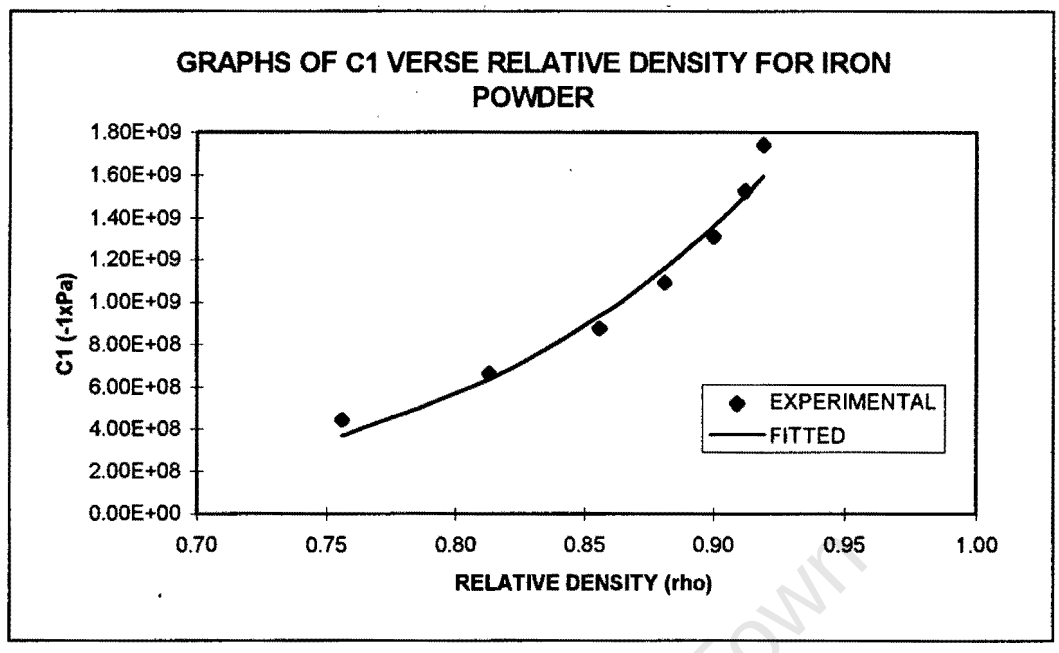

Figure E.3: C_1 curve for iron powder.

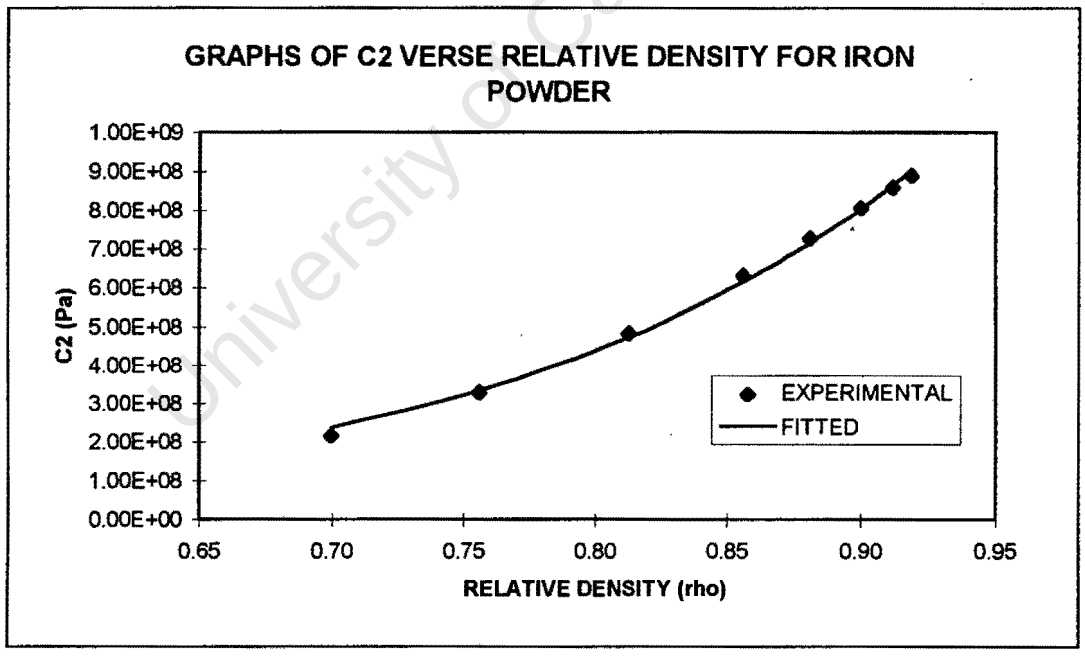

Figure E.4: C_2 curve for iron powder.

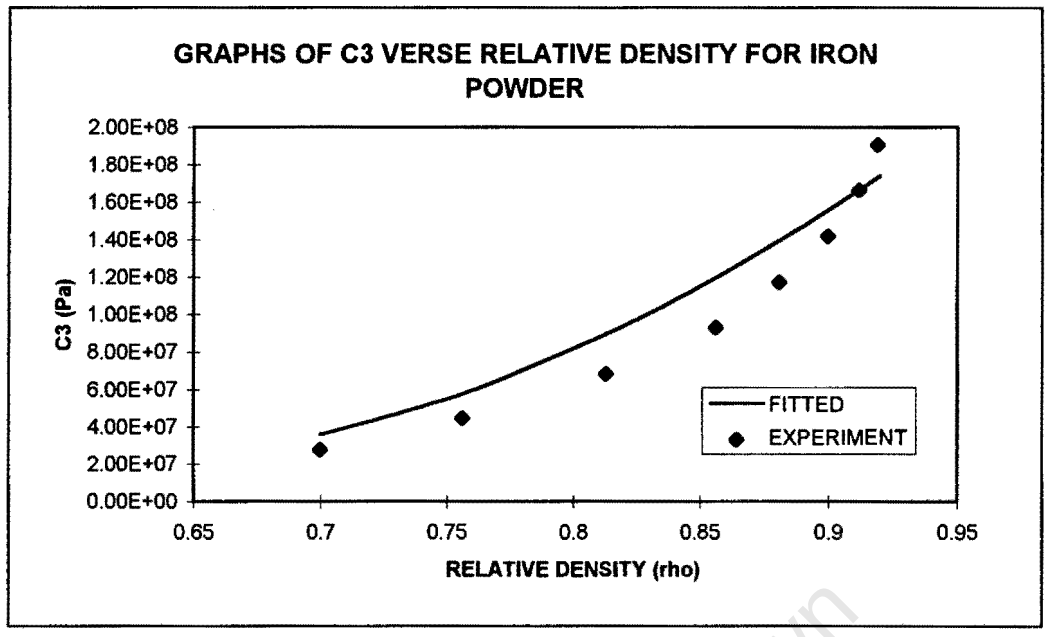

Figure E.5: C_3 curve for iron powder.

APPENDIX F

ABAQUS USER-MATERIAL SUBROUTINE CODE LISTING

```
subroutine UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
    & RPL,DDSDDT,DRPLDE,DRPLDT,
    & STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
    & NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
    & CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
c.....
c
c    ABAQUS User-Material Subroutine: UMAT
c    ------------------------------------
c
c    Implementation of a New Constitutive Model for the Finite Element
c    Analysis of Metal Powder Compaction based on work done by T.J
c    Watson and J.A. Wert (1993) on aluminium powder
c
c    The model uses shear and cap yield surfaces with density
c    evolution using a von Mises flow rule on the shear surface
c    and an associative flow rule on the cap
c
c--- Material Properties:
c    Properties Array:
c
c    PROPS(1)=mu1 or fully dense shear modulus
c    PROPS(2)=mu2 or fully dense bulk modulus
c    PROPS(3)=lam1 or fully dense poisons ratio
c    PROPS(4)=lam2 or 0.d0
c    PROPS(5)=a
c    PROPS(6)=b
c    PROPS(7)=c
c    PROPS(8)=d
c    PROPS(9)=f
c    PROPS(10)=RHO0
c    PROPS(11)=elastic moduli indicator(1=Watson(aluminium),-1=General)
c    PROPS(12)=g
c    PROPS(13)=h
c    PROPS(14)=j
c    PROPS(15)=k
c
c--- Solution Dependent State Variable:
c    NB: 3D PROBLEMS  - *DEPVAR=15
c        AXISYMMETRIC - *DEPVAR=11
c
c    1 - NTENS elastic strains
c    NTENS+1 - 2*NTENS plastic strains
c    2*NTENS+1 volumetric plastic strain
c    2*NTENS+2 relative density
```

```
c    2*NTENS+3 volumetric stress
c.....
      integer NDI,NSHR,NTENS,NSTATV,NPROPS,NOEL,NPT,LAYER
      integer KSPT,KSTEP,KINC
      real*8 STRESS(NTENS),STATEV(NSTATV),DDSDDE(NTENS,NTENS)
      real*8 SSE,SPD,SCD,RPL,DDSDDT(NTENS),DRPLDE(NTENS),DRPLDT
      real*8 PNEWDT,STRAN(NTENS),DSTRAN(NTENS),TIME(2),DTIME
      real*8 TEMP,DTEMP,PREDEF(1),DPRED(1),PROPS(NPROPS),COORDS(3)
      real*8 DROT(3,3),CELENT,DFGRD0(3,3),DFGRD1(3,3)
      character*8 CMNAME
c
      real*8 DVSTRE(7), DVSTRA(7)
      real*8 YF,YIELD,YIELDCAP,DENSITY,S
      real*8 VOLSTRE,EPVOL,RHO,LAMBDA,C1
c
c
c---  set STATEV(2*NTENS+2) to RHO0 for first inc
      STATEV(2*NTENS+2)=DENSITY(PROPS,NPROPS,STATEV(2*NTENS+1))
      RHO=STATEV(2*NTENS+2)
      if(RHO.ge.(.9999)) then
       RHO=1.D0
      end if
c
c---  set STATEV(2*NTENS+3) to VOLSTRE0 for first inc
      STATEV(2*NTENS+3)=(STRESS(1)+STRESS(2)+STRESS(3))/(3.D0)
      VOLSTRE=STATEV(2*NTENS+3)
c---  set EPVOL to EPVOL0
      EPVOL=STATEV(2*NTENS+1)
c
c---  calculate the change in deviatoric strain
      call STRADV(NTENS,DSTRAN,DVSTRA)
c
c---  calculate the elastic predictor
      call ESTRES(NTENS,NPROPS,DVSTRA,PROPS,STRESS,DVSTRE,
     &          STATEV(2*NTENS+2))
c
c---  check for active yield surface
      if((ABS(DVSTRE(NTENS+1)).ge.ABS((C1(NPROPS,PROPS,RHO))/3.)).and.
     &  (RHO.lt.(.9999))) then
c---   check for yield on cap
       YF=YIELDCAP(NTENS,NPROPS,DVSTRE,PROPS,STATEV(2*NTENS+2))
       if(YF.ge.(1.D-3)) then
c---    plastic newton iteration
        call ITCAP(PROPS,NPROPS,RHO,NTENS,DVSTRE,EPVOL,
     &           S,VOLSTRE,LAMBDA,PNEWDT)
c---    check for convergence
        if(PNEWDT.lt.(1.)) GOTO 100
c---    update deviatoric stress
        call STRUPCAP(NTENS,NPROPS,DVSTRE,PROPS,STATEV(2*NTENS+2),
     &             VOLSTRE,LAMBDA)
c---    calculate final stress
        call FSTRES(NDI,NTENS,DVSTRE,STRESS)
c---    calculate tangent modulus
        call TANCAP(NDI,NTENS,NPROPS,PROPS,LAMBDA,S,RHO,VOLSTRE,
     &           DVSTRE,DDSDDE)
c---    update state variables
        call STATCAP(NPROPS,PROPS,NSTATV,STATEV,NTENS,DVSTRE,
     &           LAMBDA,RHO,EPVOL,VOLSTRE,DSTRAN,NDI)
```

```
      else
c---    calculate final elastic stress
        call FSTRES(NDI,NTENS,DVSTRE,STRESS)
c---    calculate elastic modulus
        call JACMAT(PROPS,NPROPS,STATEV(2*NTENS+2),NDI,NTENS,DDSDDE)
c---    set arbitary values
        VOLSTRE=DVSTRE(NTENS+1)
        LAMBDA=0.D0
c---    update state variables
        call STATCAP(NPROPS,PROPS,NSTATV,STATEV,NTENS,DVSTRE,
     &            LAMBDA,RHO,EPVOL,VOLSTRE,DSTRAN,NDI)
       end if
      else if((ABS(DVSTRE(NTENS+1)).lt.ABS(C1(NPROPS,PROPS,RHO)/3.)).or.
     &     (RHO.ge.(.9999))) then
c---   check for yield on shear surface
       YF=YIELD(NTENS,NPROPS,DVSTRE,PROPS,STATEV(2*NTENS+2))
       if(YF.ge.(1.D-3)) then
c---    plastic newton iteration
        call ITMISES(PROPS,NPROPS,RHO,NTENS,DVSTRE,S,VOLSTRE,LAMBDA,
     &            PNEWDT)
c---    check for convergence
        if(PNEWDT.lt.(1.)) GOTO 100
c---    update deviatoric stress
        call STREUP(NTENS,NPROPS,DVSTRE,PROPS,STATEV(2*NTENS+2),S,
     &            VOLSTRE,LAMBDA)
c---    calculate final stress
        call FSTRES(NDI,NTENS,DVSTRE,STRESS)
c---    calculate tangent modulus
        call TANMOD(NDI,NTENS,NPROPS,PROPS,LAMBDA,S,RHO,VOLSTRE,
     &            DVSTRE,DDSDDE)
c---    update state variables
        call STATUP(NPROPS,PROPS,NSTATV,STATEV,NTENS,DVSTRE,S,
     &            LAMBDA,RHO,EPVOL,VOLSTRE,DSTRAN,NDI)
       else
c---    calculate final elastic stress
        call FSTRES(NDI,NTENS,DVSTRE,STRESS)
c---    calculate elastic modulus
        call JACMAT(PROPS,NPROPS,STATEV(2*NTENS+2),NDI,NTENS,DDSDDE)
c---    set arbitary values
        VOLSTRE=DVSTRE(NTENS+1)
        LAMBDA=0.D0
        S=1.D0
c---    update state variables
        call STATUP(NPROPS,PROPS,NSTATV,STATEV,NTENS,DVSTRE,S,
     &            LAMBDA,RHO,EPVOL,VOLSTRE,DSTRAN,NDI)
       end if
      end if
c
100   continue
      return
      end
c
      subroutine STRADV(NTENS,STRAIN,DVSTRA)
c.....
c
c     Purpose: To convert a strain vector
c              to a deviatoric strain vector
c
```

```
c i  NTENS - array length indictor
c i  STRAIN -  strain vector
c o  DVSTRA - deviatoric strain vector
c.....
c
      integer NTENS
      real*8 STRAIN(NTENS),DVSTRA(NTENS+1)
c
      DVSTRA(1)=(2./3.)*STRAIN(1)-(STRAIN(2)+STRAIN(3))/3.
      DVSTRA(2)=(2./3.)*STRAIN(2)-(STRAIN(1)+STRAIN(3))/3.
      DVSTRA(3)=(2./3.)*STRAIN(3)-(STRAIN(1)+STRAIN(2))/3.
      DVSTRA(4)=(1./2.)*STRAIN(4)
      DVSTRA(NTENS+1)=STRAIN(1)+STRAIN(2)+STRAIN(3)
      if(NTENS.eq.6) then
       DVSTRA(5)=(1./2.)*STRAIN(5)
       DVSTRA(6)=(1./2.)*STRAIN(6)
      end if
c
      return
      end
c
      subroutine ESTRES(NTENS,NPROPS,DVSTRA,PROPS,STRESS,DVSTRE,RHO)
c.....
c
c     Purpose: To calculate the elastic predictor
c
c i  NTENS - array length indictor
c i  DVSTRA - deviatoric strain vector
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  STRESS - stress vector
c o  DVSTRE - deviatoric stress vector
c i  RHO - relative density
c.....
c
      integer NTENS,NPROPS
      real*8 DVSTRA(NTENS+1),DVSTRE(NTENS+1),PROPS(NPROPS),GMOD,KMOD
      real*8 STRESS(NTENS),RHO
c
      do i=1,NTENS
       DVSTRE(i)=2*GMOD(NPROPS,PROPS,RHO)*DVSTRA(i)
      end do
      DVSTRE(NTENS+1)=KMOD(NPROPS,PROPS,RHO)*DVSTRA(NTENS+1)
c
      DVSTRE(1)=DVSTRE(1)+(2./3.)*STRESS(1)-(STRESS(2)+STRESS(3))/3.
      DVSTRE(2)=DVSTRE(2)+(2./3.)*STRESS(2)-(STRESS(1)+STRESS(3))/3.
      DVSTRE(3)=DVSTRE(3)+(2./3.)*STRESS(3)-(STRESS(1)+STRESS(2))/3.
      DVSTRE(4)=DVSTRE(4)+STRESS(4)
      DVSTRE(NTENS+1)=DVSTRE(NTENS+1)+(STRESS(1)+STRESS(2)+STRESS(3))/3.
      if(NTENS.eq.6) then
       DVSTRE(5)=DVSTRE(5)+STRESS(5)
       DVSTRE(6)=DVSTRE(6)+STRESS(6)
      end if
c
      return
      end
c
      subroutine JACMAT(PROPS,NPROPS,RHO,NDI,NTENS,DDSDDE)
```

```
c.....
c
c    Purpose: To calculate the elastic modulus
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c i  NDI - number of direct stress components
c i  NTENS - array length indictor
c o  DDSDDE - Jacobian matrix of constitutive model
c.....
      integer i,j,k,l,m,NDI,NTENS,NPROPS
      real*8 DDSDDE(NTENS,NTENS),RHO,GMOD,KMOD
      real*8 PROPS(NPROPS)
c

      do i=1,NTENS
        do j=1,NTENS
          DDSDDE(i,j)=0.0
        end do
      end do
c
      do k=1,NDI
        do l=1,NDI
          DDSDDE(l,k)=KMOD(NPROPS,PROPS,RHO)
     &              -(2./3.)*GMOD(NPROPS,PROPS,RHO)
        end do
        DDSDDE(k,k)=KMOD(NPROPS,PROPS,RHO)
     &            +(4./3.)*GMOD(NPROPS,PROPS,RHO)
      end do
c
      do m=NDI+1,NTENS
        DDSDDE(m,m)=GMOD(NPROPS,PROPS,RHO)
      end do
c
      return
      end
c
      subroutine ITMISES(PROPS,NPROPS,RHO,NTENS,DVSTRE,S,VOLSTRE,
     &                   LAMBDA,PNEWDT)
c.....
c
c    Purpose: To iterate on lambda and update stress
c             on the shear yield surface
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  NTENS - array length indictor
c i  DVSTRE - deviatoric stress vector
c i  RHO - relative density
c o  S - final SQRT(J2)
c o  VOLSTRE - final volumetric strain
c o  LAMBDA - plastic multiplier
c i/o PNEWDT - time inc. ratio
c.....
c
      integer NPROPS,NTENS,i,NUM
      real*8 PROPS(NPROPS),DVSTRE(NTENS+1)
```

```
      real*8 RHO,GMOD,ALPHA,SECINV,KAPPA
      real*8 S,LAMBDA,VOLSTRE,YF,PNEWDT,YFN,YIELD
c
c---  initial values
      LAMBDA=0.D0
      NUM=0
      YFN=ABS(YIELD(NTENS,NPROPS,DVSTRE,PROPS,RHO))
      S=SECINV(NTENS,DVSTRE)
      VOLSTRE=DVSTRE(NTENS+1)
c
      do i=1,20
       LAMBDA=LAMBDA-(S+ALPHA(NPROPS,PROPS,RHO)*VOLSTRE-
     &       KAPPA(NPROPS,PROPS,RHO))/(-GMOD(NPROPS,PROPS,RHO))
        S=SECINV(NTENS,DVSTRE)-GMOD(NPROPS,PROPS,RHO)*LAMBDA
        YF=(S+ALPHA(NPROPS,PROPS,RHO)*VOLSTRE-KAPPA(NPROPS,PROPS,RHO))
        if(ABS(YF).gt.YFN)then
         NUM=NUM+1
         if(NUM.eq.3) then
          write(6,10)
10         format(//,30X,'***WARNING - PLASTICITY ALGORITHM SOLUTION ',
     &         'DIVERGED FOR 3 ITERATIONS, SMALLER INC. TRIED')
          PNEWDT=.75
          GOTO 100
         end if
        end if
        YFN=ABS(YF)
        if((ABS(YF)).lt.(1.D-3)) GOTO 100
      end do
c
      write(6,20)
20    format(//,30X,'***WARNING - PLASTICITY ALGORITHM DID NOT ',
     &      'CONVERGE AFTER 20 ITERATIONS, SMALLER INC. TRIED')
      PNEWDT=.75
100   continue
      return
      end
c
      subroutine STREUP(NTENS,NPROPS,DVSTRE,PROPS,RHO,S,VOLSTRE,
     &              LAMBDA)
c.....
c
c     Purpose: To update the deviatoric stress if
c              plasticity has occurred
c
c i   NTENS - array length indictor
c i/o DVSTRE - deviatoric stress vector
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c i   S - SQRT(J2)
c i   VOLSTRE - volumetric stress
c i   LAMBDA - plastic multiplier
c.....
c
      integer NTENS,NPROPS
      real*8 DVSTRE(NTENS+1),PROPS(NPROPS)
      real*8 GMOD,LAMBDA,RHO,S,VOLSTRE
c
```

```
      do i=1,NTENS
       DVSTRE(i)=DVSTRE(i)/(1+((GMOD(NPROPS,PROPS,RHO)*LAMBDA)/S))
      end do
      DVSTRE(NTENS+1)=VOLSTRE
c
      return
      end
c
      subroutine FSTRES(NDI,NTENS,DVSTRE,STRESS)
c.....
c
c     Purpose: To convert a deviatoric stress vector
c              to a stress vector
c
c i   NDI - number of direct stress components
c i   NTENS - array length indictor
c i   DVSTRE - deviatoric stress vector
c o   STRESS - stress vector
c.....
c
      integer NTENS
      real*8 STRESS(NTENS),DVSTRE(NTENS+1)
c
      do i=1,NDI
       STRESS(i)=DVSTRE(i)+DVSTRE(NTENS+1)
      end do
c
      do j=NDI+1,NTENS
       STRESS(j)=DVSTRE(j)
      end do
c
      return
      end
c
      subroutine TANMOD(NDI,NTENS,NPROPS,PROPS,LAMBDA,S,RHO,VOLSTRE,
     &                  DVSTRE,DDSDDE)
c.....
c
c     Purpose: To calc the tangent modulus for
c              the shear yield surface
c
c i   NDI - number of direct stress components
c i   NTENS - array length indictor
c i   DVSTRE - deviatoric stress vector
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c i   S - SQRT(J2)
c i   VOLSTRE - volumetric stress
c i   LAMBDA - plastic multiplier
c o   DDSDDE - tangent modulus
c.....
c
      integer NDI,NTENS,NPROPS,i,j,k,l,m
      real*8 DVSTRE(NTENS+1),PROPS(NPROPS)
      real*8 DDSDDE(NTENS,NTENS),DTOTAL(7,7)
      real*8 CORSTRE(6),GMOD,LAMBDA,S,RHO,VOLSTRE,CC,KMOD,ALPHA
      real*8 sumk,sumj,termj,termk,CS3D,C3D,CSAX,CAX
```

```
c
      do i=1,NDI
       CORSTRE(i)=DVSTRE(i)
      end do
      do i=NDI+1,NTENS
       CORSTRE(i)=2*DVSTRE(i)
      end do
c
      do i=1,7
       do j=1,7
        DTOTAL(i,j)=0.D0
       end do
      end do
c
      do m=1,NTENS
       DTOTAL(m,m)=2*GMOD(NPROPS,PROPS,RHO)/(1+(GMOD(NPROPS,PROPS,RHO)*
     &          LAMBDA)/S)
      end do
c
      do i=1,NTENS
       do j=1,NTENS
        DTOTAL(i,j)=DTOTAL(i,j)+((LAMBDA*(GMOD(NPROPS,PROPS,RHO)**2)
     &          /((S**3)*(1+(GMOD(NPROPS,PROPS,RHO)*LAMBDA)/S)))
     &          -((GMOD(NPROPS,PROPS,RHO)**2)/
     &          (CC(PROPS,NPROPS,RHO,VOLSTRE)*(S**2))))
     &          *DVSTRE(i)*CORSTRE(j)
       end do
       DTOTAL(i,NTENS+1)=(-ALPHA(NPROPS,PROPS,RHO)
     &            *KMOD(NPROPS,PROPS,RHO)
     &            *GMOD(NPROPS,PROPS,RHO)
     &            /(CC(PROPS,NPROPS,RHO,VOLSTRE)*S))*DVSTRE(i)
      end do
c
      DTOTAL(NTENS+1,NTENS+1)=KMOD(NPROPS,PROPS,RHO)
c
c--- deviatoric tangent modulus transformation
c
      do i=1,NTENS
       do l=1,NTENS
        sumj=0.
        do j=1,NTENS+1
         sumk=0.
         do k=1,NTENS+1
          if(NTENS.eq.6) then
           termk=DTOTAL(j,k)*CS3D(k,l)
          else if(NTENS.eq.4) then
           termk=DTOTAL(j,k)*CSAX(k,l)
          end if
          sumk=sumk+termk
         end do
         if(NTENS.eq.6) then
          termj=C3D(i,j)*sumk
         else if(NTENS.eq.4) then
          termj=CAX(i,j)*sumk
         end if
         sumj=sumj+termj
        end do
        DDSDDE(i,l)=sumj
```

```
      end do
     end do
c
     return
     end
c
     subroutine STATUP(NPROPS,PROPS,NSTATV,STATEV,NTENS,DVSTRE,S,
    &              LAMBDA,RHO,EPVOL,VOLSTRE,DSTRAN,NDI)
c.....
c
c    Purpose: To update the state variables for the
c           shear yield surface
c
c i  NTENS - array length indictor
c i  DVSTRE - deviatoric stress vector
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  NSTATV - number of state variables
c i/o STATEV - state variables
c i  RHO - relative density
c i  S - SQRT(J2)
c i  LAMBDA - plastic multiplier
c i  EPVOL - volumetric plastic strain
c i  VOLSTRE - volumetic strain
c i  DSTRAN - strain increment
c i  NDI - number of direct strains
c.....
c
     integer NTENS,NPROPS,NSTATV,NDI,i,j,k
     real*8 DVSTRE(NTENS+1),PROPS(NPROPS),STATEV(NSTATV),DSTRAN(NTENS)
     real*8 DDVEP(7),DEP(6)
     real*8 ALPHA,LAMBDA,RHO,S,EPVOL,VOLSTRE
c
c--- DDVEP - inc in deviatoric plastic strain
     do i=1,NTENS
      DDVEP(i)=LAMBDA*DVSTRE(i)/(2*S)
     end do
      DDVEP(NTENS+1)=0.D0
c
c--- DEP - inc in plastic strain
     do j=1,NDI
      DEP(j)=DDVEP(j)+DDVEP(NTENS+1)/3
      STATEV(j)=STATEV(j)+DSTRAN(j)-DEP(j)
      STATEV(j+NTENS)=STATEV(j+NTENS)+DEP(j)
     end do
     do k=NDI+1,NTENS
      DEP(k)=2*DDVEP(k)
      STATEV(k)=STATEV(k)+DSTRAN(k)-DEP(k)
      STATEV(k+NTENS)=STATEV(k+NTENS)+DEP(k)
     end do
c
     STATEV(2*NTENS+1)=EPVOL
     STATEV(2*NTENS+2)=RHO
     STATEV(2*NTENS+3)=VOLSTRE
c
     return
     end
c
```

```
      real*8 function GMOD(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: To calculate the shear modulus
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO
c
      if(int(PROPS(11)).eq.1) then
c---    aluminium test data
        GMOD=PROPS(1)*DEXP(PROPS(2)*RHO)
      else if(int(PROPS(11)).eq.-1) then
c---    general shear modulus
        GMOD=(PROPS(1)*(RHO**2))/(1+((11.D0-19.D0*PROPS(3))*(1-RHO))/
     &       (4.D0*(1+PROPS(3))))
c
      end if
c
      return
      end
c
      real*8 function KMOD(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: To calculate the bulk modulus
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO
c
      if(int(PROPS(11)).eq.1) then
c---    aluminium test data
        KMOD=PROPS(3)*DEXP(PROPS(4)*RHO)+
     &       (2.D0/3.D0)*(PROPS(1)*DEXP(PROPS(2)*RHO))
      else if(int(PROPS(11)).eq.-1) then
c---    general bulk modulus
        KMOD=(PROPS(2)*(RHO**2))/(1+((1+PROPS(3))*(1-RHO))/
     &       (2.D0*(1-2.D0*PROPS(3))))
c
      end if
c
      return
      end
c
      real*8 function ALPHA(NPROPS,PROPS,RHO)
c.....
c
```

```
c    Purpose: To calculate the slope of shear
c             yield surface
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c.....
c
     integer NPROPS
     real*8 PROPS(NPROPS)
     real*8 RHO
c
     if(RHO.ge.(1.)) then
       ALPHA=0.D0
     else if(RHO.lt.(1.)) then
       ALPHA=PROPS(5)+PROPS(6)*RHO+PROPS(7)*(RHO**2)
     end if
c
     return
     end
c
     real*8 function KAPPA(NPROPS,PROPS,RHO)
c.....
c
c    Purpose: To calculate the intercept of the
c             shear yield surface
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c.....
c
     integer NPROPS
     real*8 PROPS(NPROPS)
     real*8 RHO
c
     KAPPA=PROPS(8)*(RHO**(PROPS(9)))
c
     return
     end
c
     real*8 function A(NPROPS,PROPS,RHO)
c.....
c
c    Purpose: A=d(ALPHA)/d(vol plastic strain)
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c.....
c
     integer NPROPS
     real*8 PROPS(NPROPS)
     real*8 RHO
c
     A=-PROPS(6)*RHO-2.D0*PROPS(7)*(RHO**2)
c
     return
```

```
      end
c
      real*8 function B(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: B=d(KAPPA)/d(vol plastic strain)
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO
c
      B=-PROPS(8)*PROPS(9)*(RHO**(PROPS(9)))
c
      return
      end
c
      real*8 function SECINV(NTENS,DVSTRE)
c.....
c
c     Purpose: To calculate the square root of J2
c
c i   NTENS - array length indictor
c i   DVSTRE - deviatoric stress vector
c.....
c
      integer NTENS
      real*8 DVSTRE(NTENS+1)
c
      if(NTENS.eq.6) then
        SECINV=DSQRT((1./2.)*(DVSTRE(1)**2+DVSTRE(2)**2+DVSTRE(3)**2)
     &       +DVSTRE(4)**2+DVSTRE(5)**2+DVSTRE(6)**2)
      else if(NTENS.eq.4) then
        SECINV=DSQRT((1./2.)*(DVSTRE(1)**2+DVSTRE(2)**2+DVSTRE(3)**2)
     &       +DVSTRE(4)**2)
      end if
c
      return
      end
c
      real*8 function YIELD(NTENS,NPROPS,DVSTRE,PROPS,RHO)
c.....
c
c     Purpose: To calculate the value of the shear yield func
c
c i   NTENS - array length indictor
c i   DVSTRE - deviatoric stress vector
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NTENS,NPROPS
      real*8 DVSTRE(NTENS+1),PROPS(NPROPS)
```

```
      real*8 SECINV,ALPHA,KAPPA,RHO
c
c
      YIELD=SECINV(NTENS,DVSTRE)+ALPHA(NPROPS,PROPS,RHO)*DVSTRE(NTENS+1)
     &     -KAPPA(NPROPS,PROPS,RHO)
c
      return
      end
c
      real*8 function DENSITY(PROPS,NPROPS,EPVOL)
c.....
c
c     Purpose: To calculate the density for a given
c              volumetric plastic strain
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   EPVOL - volumetric plastic strain
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 EPVOL
c
      DENSITY=PROPS(10)*DEXP(-EPVOL)
c
      return
      end
c
      real*8 function CC(PROPS,NPROPS,RHO,VOLSTRE)
c.....
c
c     Purpose: To calculate a variable in the
c              tangent modulus for the shear yield surface
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c i   VOLSTRE - volumetric stress
c.....
c
      integer NPROPS
      real*8 RHO,VOLSTRE,PROPS(NPROPS)
      real*8 ALPHA,GMOD,KMOD,A,B
c
       CC=GMOD(NPROPS,PROPS,RHO)
c
      return
      end
c
      real*8 function CS3D(row,col)
c.....
c
c     Purpose: To define the strain to deviatoric
c              strain transformation tensor for 3D
c
c i   row - row indicator
c i   col - column indicator
```

```
c.....
      integer row,col
c
      if(row.eq.1) then
       if(col.eq.1) then
         CS3D=2./3.
       else if(col.eq.2) then
         CS3D=-1./3.
       else if(col.eq.3) then
         CS3D=-1./3.
       else if(col.eq.4) then
         CS3D=0.
       else if(col.eq.5) then
         CS3D=0.
       else if(col.eq.6) then
         CS3D=0.
       end if
      else if(row.eq.2) then
       if(col.eq.1) then
         CS3D=-1./3.
       else if(col.eq.2) then
         CS3D=2./3.
       else if(col.eq.3) then
         CS3D=-1./3.
       else if(col.eq.4) then
         CS3D=0.
       else if(col.eq.5) then
         CS3D=0.
       else if(col.eq.6) then
         CS3D=0.
       end if
      else if(row.eq.3) then
       if(col.eq.1) then
         CS3D=-1./3.
       else if(col.eq.2) then
         CS3D=-1./3.
       else if(col.eq.3) then
         CS3D=2./3.
       else if(col.eq.4) then
         CS3D=0.
       else if(col.eq.5) then
         CS3D=0.
       else if(col.eq.6) then
         CS3D=0.
       end if
      else if(row.eq.4) then
       if(col.eq.1) then
         CS3D=0.
       else if(col.eq.2) then
         CS3D=0.
       else if(col.eq.3) then
         CS3D=0.
       else if(col.eq.4) then
         CS3D=1./2.
       else if(col.eq.5) then
         CS3D=0.
       else if(col.eq.6) then
         CS3D=0.
```

```
      end if
     else if(row.eq.5) then
      if(col.eq.1) then
        CS3D=0.
      else if(col.eq.2) then
        CS3D=0.
      else if(col.eq.3) then
        CS3D=0.
      else if(col.eq.4) then
        CS3D=0.
      else if(col.eq.5) then
        CS3D=1./2.
      else if(col.eq.6) then
        CS3D=0.
      end if
     else if(row.eq.6) then
      if(col.eq.1) then
        CS3D=0.
      else if(col.eq.2) then
        CS3D=0.
      else if(col.eq.3) then
        CS3D=0.
      else if(col.eq.4) then
        CS3D=0.
      else if(col.eq.5) then
        CS3D=0.
      else if(col.eq.6) then
        CS3D=1./2.
      end if
     else if(row.eq.7) then
      if(col.eq.1) then
        CS3D=1.
      else if(col.eq.2) then
        CS3D=1.
      else if(col.eq.3) then
        CS3D=1.
      else if(col.eq.4) then
        CS3D=0.
      else if(col.eq.5) then
        CS3D=0.
      else if(col.eq.6) then
        CS3D=0.
      end if
     end if
c
     return
     end
c
     real*8 function C3D(row,col)
c.....
c
c    Purpose: To define the deviatoric stress to
c            stress transformation tensor for 3D
c
c i  row - row indicator
c i  col - column indicator
c.....
     integer row,col
```

```
c
      if(row.eq.1) then
       if(col.eq.1) then
         C3D=1.
       else if(col.eq.2) then
         C3D=0.
       else if(col.eq.3) then
         C3D=0.
       else if(col.eq.4) then
         C3D=0.
       else if(col.eq.5) then
         C3D=0.
       else if(col.eq.6) then
         C3D=0.
       else if(col.eq.7) then
         C3D=1.
       end if
      else if(row.eq.2) then
       if(col.eq.1) then
         C3D=0.
       else if(col.eq.2) then
         C3D=1.
       else if(col.eq.3) then
         C3D=0.
       else if(col.eq.4) then
         C3D=0.
       else if(col.eq.5) then
         C3D=0.
       else if(col.eq.6) then
         C3D=0.
       else if(col.eq.7) then
         C3D=1.
       end if
      else if(row.eq.3) then
       if(col.eq.1) then
         C3D=0.
       else if(col.eq.2) then
         C3D=0.
       else if(col.eq.3) then
         C3D=1.
       else if(col.eq.4) then
         C3D=0.
       else if(col.eq.5) then
         C3D=0.
       else if(col.eq.6) then
         C3D=0.
       else if(col.eq.7) then
         C3D=1.
       end if
      else if(row.eq.4) then
       if(col.eq.1) then
         C3D=0.
       else if(col.eq.2) then
         C3D=0.
       else if(col.eq.3) then
         C3D=0.
       else if(col.eq.4) then
         C3D=1.
```

```
      else if(col.eq.5) then
        C3D=0.
      else if(col.eq.6) then
        C3D=0.
      else if(col.eq.7) then
        C3D=0.
      end if
     else if(row.eq.5) then
      if(col.eq.1) then
        C3D=0.
      else if(col.eq.2) then
        C3D=0.
      else if(col.eq.3) then
        C3D=0.
      else if(col.eq.4) then
        C3D=0.
      else if(col.eq.5) then
        C3D=1.
      else if(col.eq.6) then
        C3D=0.
      else if(col.eq.7) then
        C3D=0.
      end if
     else if(row.eq.6) then
      if(col.eq.1) then
        C3D=0.
      else if(col.eq.2) then
        C3D=0.
      else if(col.eq.3) then
        C3D=0.
      else if(col.eq.4) then
        C3D=0.
      else if(col.eq.5) then
        C3D=0.
      else if(col.eq.6) then
        C3D=1.
      else if(col.eq.7) then
        C3D=0.
      end if
     end if
c
     return
     end
c
     real*8 function CAX(row,col)
c.....
c
c    Purpose: To define the deviatoric stress to
c             stress transformation tensor for
c             axisymmetric
c
c i  row - row indicator
c i  col - column indicator
c.....
     integer row,col
c
     if(row.eq.1) then
      if(col.eq.1) then
```

```
        CAX=1.
      else if(col.eq.2) then
        CAX=0.
      else if(col.eq.3) then
        CAX=0.
      else if(col.eq.4) then
        CAX=0.
      else if(col.eq.5) then
        CAX=1.
      end if
    else if(row.eq.2) then
      if(col.eq.1) then
        CAX=0.
      else if(col.eq.2) then
        CAX=1.
      else if(col.eq.3) then
        CAX=0.
      else if(col.eq.4) then
        CAX=0.
      else if(col.eq.5) then
        CAX=1.
      end if
    else if(row.eq.3) then
      if(col.eq.1) then
        CAX=0.
      else if(col.eq.2) then
        CAX=0.
      else if(col.eq.3) then
        CAX=1.
      else if(col.eq.4) then
        CAX=0.
      else if(col.eq.5) then
        CAX=1.
      end if
    else if(row.eq.4) then
      if(col.eq.1) then
        CAX=0.
      else if(col.eq.2) then
        CAX=0.
      else if(col.eq.3) then
        CAX=0.
      else if(col.eq.4) then
        CAX=1.
      else if(col.eq.5) then
        CAX=0.
      end if
    end if
c
    return
    end
c
    real*8 function CSAX(row,col)
c.....
c
c   Purpose: To define the strain to deviatoric
c            strain transformation tensor for
c            axisymmetric
c
```

```
c i  row - row indicator
c i  col - column indicator
c.....................................................
      integer row,col
c
      if(row.eq.1) then
       if(col.eq.1) then
         CSAX=2./3.
       else if(col.eq.2) then
         CSAX=-1./3.
       else if(col.eq.3) then
         CSAX=-1./3.
       else if(col.eq.4) then
         CSAX=0.
       end if
      else if(row.eq.2) then
       if(col.eq.1) then
         CSAX=-1./3.
       else if(col.eq.2) then
         CSAX=2./3.
       else if(col.eq.3) then
         CSAX=-1./3.
       else if(col.eq.4) then
         CSAX=0.
       end if
      else if(row.eq.3) then
       if(col.eq.1) then
         CSAX=-1./3.
       else if(col.eq.2) then
         CSAX=-1./3.
       else if(col.eq.3) then
         CSAX=2./3.
       else if(col.eq.4) then
         CSAX=0.
       end if
      else if(row.eq.4) then
       if(col.eq.1) then
         CSAX=0.
       else if(col.eq.2) then
         CSAX=0.
       else if(col.eq.3) then
         CSAX=0.
       else if(col.eq.4) then
         CSAX=1./2.
       end if
      else if(row.eq.5) then
       if(col.eq.1) then
         CSAX=1.
       else if(col.eq.2) then
         CSAX=1.
       else if(col.eq.3) then
         CSAX=1.
       else if(col.eq.4) then
         CSAX=0.
       end if
      end if
c
      return
```

```
      end
c
      subroutine ITCAP(PROPS,NPROPS,RHO,NTENS,DVSTRE,EPVOL,
     &                 S,VOLSTRE,LAMBDA,PNEWDT)
c.....
c
c     Purpose: To iterate on lambda and
c              update stress for cap yield surface
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  NTENS - array length indictor
c i  DVSTRE - deviatoric stress vector
c i  RHO - relative density
c i  EPVOL - volumetric plastic strain
c o  S - final SQRT(J2)
c o  VOLSTRE - final volumetric strain
c o  LAMBDA - plastic multiplier
c i/o PNEWDT - time inc. ratio
c.....
c
      integer NPROPS,NTENS,i,j,NUM
      real*8 PROPS(NPROPS),DVSTRE(NTENS+1),PNEWDT,DERF,DNLAMBDA
      real*8 EPVOL,RHO,GMOD,KMOD,DENSITY,SECINV,YIELDCAP,VOLSTRE1
      real*8 RHO0,S,LAMBDA,VOLSTRE,YF,YFN,C1,C2,C3,E,F,M,N,DNL
c
c---  initial values
      LAMBDA=0.D0
      NUM=0
      YFN=ABS(YIELDCAP(NTENS,NPROPS,DVSTRE,PROPS,RHO))
      RHO0=RHO
      EPVOL0=EPVOL
      S=SECINV(NTENS,DVSTRE)
      VOLSTRE=DVSTRE(NTENS+1)
      VOLSTRE1=VOLSTRE
      DNLAMBDA=DNL(NPROPS,PROPS,RHO,RHO0,VOLSTRE,LAMBDA)
c
      do i=1,20
c---    derivative of yield func with respect to lambda
       DERF=(((-4.D0*(S**2)*
     &      GMOD(NPROPS,PROPS,RHO0))/(1+2.D0*GMOD(NPROPS,PROPS,RHO0)
     &      *LAMBDA))-2.D0*((C3(NPROPS,PROPS,RHO))
     &      /(C2(NPROPS,PROPS,RHO)**2))*((C2(NPROPS,PROPS,RHO)**2)
     &      -(3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))**2)*M(NPROPS,PROPS,RHO)
     &      *(N(NPROPS,PROPS,RHO,VOLSTRE)+LAMBDA*
     &      DNLAMBDA)+2.D0*
     &      ((C3(NPROPS,PROPS,RHO)**2)
     &      /(C2(NPROPS,PROPS,RHO)**3))*((C2(NPROPS,PROPS,RHO)**2)
     &      -(3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))**2)*F(NPROPS,PROPS,RHO)
     &      *(N(NPROPS,PROPS,RHO,VOLSTRE)+LAMBDA*
     &      DNLAMBDA)-
     &      ((C3(NPROPS,PROPS,RHO)**2)/
     &      (C2(NPROPS,PROPS,RHO)**2))*(2.D0*(C2(NPROPS,PROPS,RHO))
     &      *F(NPROPS,PROPS,RHO)*(N(NPROPS,PROPS,RHO,VOLSTRE)+LAMBDA*
     &      DNLAMBDA)-2.D0*
     &      (3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))*(-3.D0*
     &      KMOD(NPROPS,PROPS,RHO0)*(N(NPROPS,PROPS,RHO,VOLSTRE)+LAMBDA*
     &      DNLAMBDA)
```

```
     &     -E(NPROPS,PROPS,RHO)*(N(NPROPS,PROPS,RHO,VOLSTRE)+LAMBDA*
     &     DNLAMBDA))))
c
c---  lambda
      LAMBDA=LAMBDA-((S**2)-((C3(NPROPS,PROPS,RHO)**2)/
     &      (C2(NPROPS,PROPS,RHO)**2))*((C2(NPROPS,PROPS,RHO)**2)
     &      -(3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))**2))/DERF
c
      do j=1,10
       EPVOL=EPVOL0+N(NPROPS,PROPS,RHO,VOLSTRE)*LAMBDA
       RHO=DENSITY(PROPS,NPROPS,EPVOL)
       VOLSTRE=(DVSTRE(NTENS+1)*(C2(NPROPS,PROPS,RHO)**2)+
     &        6.D0*KMOD(NPROPS,PROPS,RHO0)*(C3(NPROPS,PROPS,RHO)**2)*
     &        LAMBDA*C1(NPROPS,PROPS,RHO))/((C2(NPROPS,PROPS,RHO)**2)
     &        +18.D0*KMOD(NPROPS,PROPS,RHO0)*LAMBDA
     &        *(C3(NPROPS,PROPS,RHO)**2))
      end do
      S=SECINV(NTENS,DVSTRE)/(1+2.D0*GMOD(NPROPS,PROPS,RHO0)*LAMBDA)
      DNLAMBDA=DNL(NPROPS,PROPS,RHO,RHO0,VOLSTRE,LAMBDA)
      YF=S**2-((C3(NPROPS,PROPS,RHO)**2)/
     &   (C2(NPROPS,PROPS,RHO)**2))*((C2(NPROPS,PROPS,RHO)**2)
     &   -(3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))**2)
      if(ABS(YF).gt.YFN)then
       NUM=NUM+1
       if(NUM.eq.3) then
        write(6,30)
30       format(//,30X,'***WARNING - PLASTICITY ALGORITHM SOLUTION ',
     &        'DIVERGED FOR 3 ITERATIONS, SMALLER INC. TRIED')
        PNEWDT=.75
        GOTO 100
       end if
      end if
      YFN=ABS(YF)
      if((ABS(YF)).lt.(1.D-3)) GOTO 100
     end do
c
     write(6,40)
40   format(//,30X,'***WARNING - PLASTICITY ALGORITHM DID NOT ',
     &     'CONVERGE AFTER 20 ITERATIONS, SMALLER INC. TRIED')
     PNEWDT=.75
100  continue
c
     if(RHO.gt.(1.D0)) then
      write(6,110)
110    format(//,30X,'***WARNING - RHO EXCEEDED FULL DENSITY ',
     &     ', SMALLER INC. TRIED')
      PNEWDT=.75
     end if
c
     return
     end
c
     subroutine STRUPCAP(NTENS,NPROPS,DVSTRE,PROPS,RHO,VOLSTRE,
     &              LAMBDA)
c.....
c
c    Purpose: To update the deviatoric stress if
c           plasticity has occurred on the cap
```

```
c
c i  NTENS - array length indictor
c i  DVSTRE - deviatoric stress vector
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c i  VOLSTRE - volumetric stress
c i  LAMBDA - plastic multiplier
c.....
c
      integer NTENS,NPROPS
      real*8 DVSTRE(NTENS+1),PROPS(NPROPS)
      real*8 GMOD,LAMBDA,RHO,VOLSTRE
c
      do i=1,NTENS
       DVSTRE(i)=DVSTRE(i)/(1+2.*GMOD(NPROPS,PROPS,RHO)*LAMBDA)
      end do
      DVSTRE(NTENS+1)=VOLSTRE
c
      return
      end
c
      subroutine TANCAP(NDI,NTENS,NPROPS,PROPS,LAMBDA,S,RHO,VOLSTRE,
     &                  DVSTRE,DDSDDE)
c.....
c
c     Purpose: To calc the tangent modulus for the cap
c              yield surface
c
c i  NDI - number of direct stress components
c i  NTENS - array length indictor
c i  DVSTRE - deviatoric stress vector
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c i  S - SQRT(J2)
c i  VOLSTRE - volumetric stress
c i  LAMBDA - plastic multiplier
c o  DDSDDE - tangent modulus
c.....
c
      integer NDI,NTENS,NPROPS,i,j,k,l,m
      real*8 DVSTRE(NTENS+1),PROPS(NPROPS)
      real*8 DDSDDE(NTENS,NTENS),DTOTAL(7,7)
      real*8 CORSTRE(6),GMOD,LAMBDA,S,RHO,VOLSTRE,CCCAP,KMOD
      real*8 sumk,sumj,termj,termk,CS3D,C3D,CSAX,CAX,X,N,T
c
      do i=1,NDI
       CORSTRE(i)=DVSTRE(i)
      end do
      do i=NDI+1,NTENS
       CORSTRE(i)=2*DVSTRE(i)
      end do
c
      do i=1,7
       do j=1,7
        DTOTAL(i,j)=0.D0
       end do
```

```
      end do
c
      do m=1,NTENS
       DTOTAL(m,m)=2*GMOD(NPROPS,PROPS,RHO)/(1+2.*
     &          GMOD(NPROPS,PROPS,RHO)*LAMBDA)
      end do
c
      do i=1,NTENS
       do j=1,NTENS
        DTOTAL(i,j)=DTOTAL(i,j)-((4.*(GMOD(NPROPS,PROPS,RHO)**2))
     &           /(((1+2.*GMOD(NPROPS,PROPS,RHO)*LAMBDA)**2)*
     &           CCCAP(NPROPS,PROPS,RHO,VOLSTRE,S,LAMBDA)))
     &           *DVSTRE(i)*CORSTRE(j)
       end do
       DTOTAL(i,NTENS+1)=((-2.*X(NPROPS,PROPS,RHO,VOLSTRE,LAMBDA)
     &             *KMOD(NPROPS,PROPS,RHO)
     &             *GMOD(NPROPS,PROPS,RHO))
     &             /(CCCAP(NPROPS,PROPS,RHO,VOLSTRE,S,LAMBDA)*
     &             (1+2.*GMOD(NPROPS,PROPS,RHO)*LAMBDA)))
     &             *DVSTRE(i)
c
       DTOTAL(NTENS+1,i)=((-2.*N(NPROPS,PROPS,RHO,VOLSTRE)
     &             *KMOD(NPROPS,PROPS,RHO)
     &             *GMOD(NPROPS,PROPS,RHO))
     &             /(CCCAP(NPROPS,PROPS,RHO,VOLSTRE,S,LAMBDA)*
     &             (1+2.*GMOD(NPROPS,PROPS,RHO)*LAMBDA)*(1+
     &             KMOD(NPROPS,PROPS,RHO)*T(NPROPS,PROPS,RHO)
     &             *LAMBDA)))*CORSTRE(i)
      end do
c
      DTOTAL(NTENS+1,NTENS+1)=(KMOD(NPROPS,PROPS,RHO)-
     &             (X(NPROPS,PROPS,RHO,VOLSTRE,LAMBDA)*
     &             N(NPROPS,PROPS,RHO,VOLSTRE)*
     &             (KMOD(NPROPS,PROPS,RHO)**2))/
     &             (CCCAP(NPROPS,PROPS,RHO,VOLSTRE,S,LAMBDA)))/
     &             (1+KMOD(NPROPS,PROPS,RHO)*
     &             T(NPROPS,PROPS,RHO)*LAMBDA)
c
c--- deviatoric tangent modulus transformation
      do i=1,NTENS
       do l=1,NTENS
        sumj=0.
        do j=1,NTENS+1
         sumk=0.
         do k=1,NTENS+1
          if(NTENS.eq.6) then
           termk=DTOTAL(j,k)*CS3D(k,l)
          else if(NTENS.eq.4) then
           termk=DTOTAL(j,k)*CSAX(k,l)
          end if
          sumk=sumk+termk
         end do
         if(NTENS.eq.6) then
          termj=C3D(i,j)*sumk
         else if(NTENS.eq.4) then
          termj=CAX(i,j)*sumk
         end if
         sumj=sumj+termj
```

```
        end do
        DDSDDE(i,l)=sumj
       end do
      end do
c
      return
      end
c
      subroutine STATCAP(NPROPS,PROPS,NSTATV,STATEV,NTENS,DVSTRE,
     &            LAMBDA,RHO,EPVOL,VOLSTRE,DSTRAN,NDI)
c.....
c
c    Purpose: To update the state variables for cap
c          yield surface
c
c i  NTENS - array length indictor
c i  DVSTRE - deviatoric stress vector
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  NSTATV - number of state variables
c i/o STATEV - state variables
c i  RHO - relative density
c i  LAMBDA - plastic multiplier
c i  EPVOL - volumetric plastic strain
c i  VOLSTRE - volumetic strain
c i  DSTRAN - strain increment
c i  NDI - number of direct strains
c.....
c
      integer NTENS,NPROPS,NSTATV,NDI,i,j,k
      real*8 DVSTRE(NTENS+1),PROPS(NPROPS),STATEV(NSTATV),DSTRAN(NTENS)
      real*8 DDVEP(7),DEP(6)
      real*8 LAMBDA,RHO,EPVOL,VOLSTRE,N
c
c---  DDVEP - inc in deviatoric plastic strain
      do i=1,NTENS
       DDVEP(i)=LAMBDA*DVSTRE(i)
      end do
c
      DDVEP(NTENS+1)=N(NPROPS,PROPS,RHO,VOLSTRE)*LAMBDA
c
c---  DEP - inc in plastic strain
      do j=1,NDI
       DEP(j)=DDVEP(j)+DDVEP(NTENS+1)/3
       STATEV(j)=STATEV(j)+DSTRAN(j)-DEP(j)
       STATEV(j+NTENS)=STATEV(j+NTENS)+DEP(j)
      end do
      do k=NDI+1,NTENS
       DEP(k)=2*DDVEP(k)
       STATEV(k)=STATEV(k)+DSTRAN(k)-DEP(k)
       STATEV(k+NTENS)=STATEV(k+NTENS)+DEP(k)
      end do
c
      STATEV(2*NTENS+1)=EPVOL
      STATEV(2*NTENS+2)=RHO
      STATEV(2*NTENS+3)=VOLSTRE
c
      return
```

```
      end
c
      real*8 function C1(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: To calculate C1 variable in the cap
c              yield surface
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO
c
      C1=PROPS(12)*(RHO**(PROPS(13)))
c
      return
      end
c
      real*8 function E(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: E=d(C1)/d(vol plastic strain)
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO
c
      E=-PROPS(12)*PROPS(13)*(RHO**(PROPS(13)))
c
      return
      end
c
      real*8 function C2(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: To calculate C2 variable in cap
c              yield surface
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO
c
       C2=PROPS(14)*DEXP(PROPS(15)*RHO)
c
```

```
      return
      end
c
      real*8 function F(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: F=d(C2)/d(vol plastic strain)
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO
c
      F=-PROPS(14)*PROPS(15)*RHO*DEXP(PROPS(15)*RHO)
c
      return
      end
c
      real*8 function C3(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: To calculate C3 variable in cap
c           yield surface
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO,ALPHA,KAPPA,C1
c
      C3=ALPHA(NPROPS,PROPS,RHO)*ABS((C1(NPROPS,PROPS,RHO))/3.D0)
     &   +KAPPA(NPROPS,PROPS,RHO)
c
      return
      end
c
      real*8 function M(NPROPS,PROPS,RHO)
c.....
c
c     Purpose: M=d(C3)/d(vol plastic strain)
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO,ALPHA,A,B,C1,E
c
      M=A(NPROPS,PROPS,RHO)*(-C1(NPROPS,PROPS,RHO)/3.D0)+
```

```
     & ALPHA(NPROPS,PROPS,RHO)
     & *(-E(NPROPS,PROPS,RHO)/3.D0)+B(NPROPS,PROPS,RHO)
c
      return
      end
c
      real*8 function N(NPROPS,PROPS,RHO,VOLSTRE)
c.....
c
c     Purpose: N=d(f)/d(volstre)
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c i   VOLSTRE - volumetric stress
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO,VOLSTRE,C1,C2,C3
c
      N=6.D0*((C3(NPROPS,PROPS,RHO)**2)/(C2(NPROPS,PROPS,RHO)**2))*
     & (3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))
c
      return
      end
c
      real*8 function DNL(NPROPS,PROPS,RHO,RHO0,VOLSTRE,LAMBDA)
c.....
c
c     Purpose: DNL=d(N)/d(LAMBDA)
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c i   VOLSTRE - volumetric stress
c i   LAMBDA - plastic multiplier
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO,RHO0,VOLSTRE,LAMBDA,C1,C2,C3,N,E,F,M,KMOD
      real*8 TOP,BOT
c
      TOP=(6.D0*M(NPROPS,PROPS,RHO)*VOLSTRE*C2(NPROPS,PROPS,RHO)-2.D0*
     &   M(NPROPS,PROPS,RHO)*C1(NPROPS,PROPS,RHO)*C2(NPROPS,PROPS,RHO)-
     &   6.D0*C3(NPROPS,PROPS,RHO)*F(NPROPS,PROPS,RHO)*VOLSTRE+2.D0*
     &   C3(NPROPS,PROPS,RHO)*F(NPROPS,PROPS,RHO)*C1(NPROPS,PROPS,RHO)-
     &   3.D0*C3(NPROPS,PROPS,RHO)*KMOD(NPROPS,PROPS,RHO0)*
     &   C2(NPROPS,PROPS,RHO)-C3(NPROPS,PROPS,RHO)*E(NPROPS,PROPS,RHO)
     &   *C2(NPROPS,PROPS,RHO))
      BOT=(-(C2(NPROPS,PROPS,RHO)**3)+36.D0*C3(NPROPS,PROPS,RHO)*
     &   M(NPROPS,PROPS,RHO)*LAMBDA*VOLSTRE*C2(NPROPS,PROPS,RHO)-12.D0*
     &   C3(NPROPS,PROPS,RHO)*M(NPROPS,PROPS,RHO)*C1(NPROPS,PROPS,RHO)*
     &   C2(NPROPS,PROPS,RHO)*LAMBDA-36.D0*(C3(NPROPS,PROPS,RHO)**2)*
     &   F(NPROPS,PROPS,RHO)*LAMBDA*VOLSTRE+12.D0*F(NPROPS,PROPS,RHO)*
     &   (C3(NPROPS,PROPS,RHO)**2)*C1(NPROPS,PROPS,RHO)*LAMBDA-18.D0*
     &   (C3(NPROPS,PROPS,RHO)**2)*KMOD(NPROPS,PROPS,RHO0)*LAMBDA*
```

```
     &   C2(NPROPS,PROPS,RHO)-6.D0*(C3(NPROPS,PROPS,RHO)**2)*LAMBDA*
     &   E(NPROPS,PROPS,RHO)*C2(NPROPS,PROPS,RHO))
c
     DNL=-6.D0*C3(NPROPS,PROPS,RHO)*N(NPROPS,PROPS,RHO,VOLSTRE)*(TOP/
     & BOT)
c
     return
     end
c
     real*8 function T(NPROPS,PROPS,RHO)
c.....
c
c    Purpose: T=d(2f)/d(volstre^2)
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c.....
c
     integer NPROPS
     real*8 PROPS(NPROPS)
     real*8 RHO,C2,C3
c
     T=18.D0*((C3(NPROPS,PROPS,RHO)**2)/(C2(NPROPS,PROPS,RHO)**2))
c
     return
     end
c
     real*8 function P(NPROPS,PROPS,RHO,VOLSTRE)
c.....
c
c    Purpose: P=d(f)/d(C1)
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c i  VOLSTRE - volumetric stress
c.....
c
     integer NPROPS
     real*8 PROPS(NPROPS)
     real*8 RHO,VOLSTRE,C1,C2,C3
c
     P=-2.D0*((C3(NPROPS,PROPS,RHO)**2)/(C2(NPROPS,PROPS,RHO)**2))*
     &  (3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))
c
     return
     end
c
     real*8 function Q(NPROPS,PROPS,RHO,VOLSTRE)
c.....
c
c    Purpose: Q=d(f)/d(C2)
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c i  VOLSTRE - volumetric stress
```

```
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO,VOLSTRE,C1,C2,C3
c
      Q=-2.D0*((C3(NPROPS,PROPS,RHO)**2)/(C2(NPROPS,PROPS,RHO)**3))*
     & (3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))**2
c
      return
      end
c
      real*8 function R(NPROPS,PROPS,RHO,VOLSTRE)
c.....
c
c     Purpose: R=d(f)/d(C3)
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c i  VOLSTRE - volumetric stress
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO,VOLSTRE,C1,C2,C3
c
      R=2.D0*((C3(NPROPS,PROPS,RHO))/(C2(NPROPS,PROPS,RHO)**2))*
     & ((3.D0*VOLSTRE-C1(NPROPS,PROPS,RHO))**2)
     & -2.D0*C3(NPROPS,PROPS,RHO)
c
      return
      end
c
      real*8 function X(NPROPS,PROPS,RHO,VOLSTRE,LAMBDA)
c.....
c
c     Purpose: calc var for cap tangent modulus
c
c i  PROPS - array containing material properties
c i  NPROPS - number of material properties
c i  RHO - relative density
c i  VOLSTRE - volumetric stress
c i  LAMBDA - plastic multiplier
c.....
c
      integer NPROPS
      real*8 PROPS(NPROPS)
      real*8 RHO,VOLSTRE,N,P,Q,R,E,F,M,T,KMOD,LAMBDA
c
      X=(N(NPROPS,PROPS,RHO,VOLSTRE)+(P(NPROPS,PROPS,RHO,VOLSTRE)*
     & E(NPROPS,PROPS,RHO)+Q(NPROPS,PROPS,RHO,VOLSTRE)*
     & F(NPROPS,PROPS,RHO)+R(NPROPS,PROPS,RHO,VOLSTRE)*
     & M(NPROPS,PROPS,RHO))*LAMBDA*T(NPROPS,PROPS,RHO))/
     & (1+KMOD(NPROPS,PROPS,RHO)*T(NPROPS,PROPS,RHO)*LAMBDA)
c
      return
      end
```

```
c
      real*8 function CCCAP(NPROPS,PROPS,RHO,VOLSTRE,S,LAMBDA)
c.....
c
c     Purpose: To calculate a variable in the
c              cap tangent modulus
c
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c i   S - SQRT(J2)
c i   LAMBDA - plastic multiplier
c i   VOLSTRE - volumetric stress
c.....
c
      integer NPROPS
      real*8 RHO,VOLSTRE,PROPS(NPROPS),S,LAMBDA
      real*8 GMOD,KMOD,P,Q,R,E,F,M,N,X
c
      CCCAP=((4.D0*(S**2)*GMOD(NPROPS,PROPS,RHO))/(1+2.D0*
     &      GMOD(NPROPS,PROPS,RHO)*LAMBDA))-
     &      (P(NPROPS,PROPS,RHO,VOLSTRE)*
     &      E(NPROPS,PROPS,RHO)+Q(NPROPS,PROPS,RHO,VOLSTRE)*
     &      F(NPROPS,PROPS,RHO)+R(NPROPS,PROPS,RHO,VOLSTRE)*
     &      M(NPROPS,PROPS,RHO))*N(NPROPS,PROPS,RHO,VOLSTRE)+
     &      X(NPROPS,PROPS,RHO,VOLSTRE,LAMBDA)*KMOD(NPROPS,PROPS,RHO)
     &      *N(NPROPS,PROPS,RHO,VOLSTRE)
c
      return
      end
c
      real*8 function YIELDCAP(NTENS,NPROPS,DVSTRE,PROPS,RHO)
c.....
c
c     Purpose: To calculate the value of the cap
c              yield function
c
c i   NTENS - array length indictor
c i   DVSTRE - deviatoric stress vector
c i   PROPS - array containing material properties
c i   NPROPS - number of material properties
c i   RHO - relative density
c.....
c
      integer NTENS,NPROPS
      real*8 DVSTRE(NTENS+1),PROPS(NPROPS)
      real*8 SECINV,RHO,C1,C2,C3
c
c
      YIELDCAP=((SECINV(NTENS,DVSTRE))**2)-((C3(NPROPS,PROPS,RHO)**2)/
     &      (C2(NPROPS,PROPS,RHO)**2))*((C2(NPROPS,PROPS,RHO)**2)
     &      -(3.D0*DVSTRE(NTENS+1)-C1(NPROPS,PROPS,RHO))**2)
c
      return
      end
```

APPENDIX G

ABAQUS INPUT FILE: HYDROSTATIC AND CONSTRAINED COMPRESSION TESTS

```
*HEADING
INPUT FILE USED FOR HYDROSTATIC AND CONSTRAINED COMPRESSION TESTS
**
**
**------------------------------------------------------------
**  Define corner nodes
**
*NODE
1,  0.,  0., 0.
2, .02,  0., 0.
3, .02, .02, 0.
4, 0., .02, 0.
5, 0., 0., .02
6, .02, 0., .02
7, .02, .02, .02
8, 0., .02, .02
**
*NSET,NSET=SIDE3
5,6,7,8
*NSET,NSET=SIDE1
2,3,6,7
*NSET,NSET=SIDE2
3,4,7,8
**
**------------------------------------------------------------
**
**  Define master element
**
*ELEMENT, TYPE=C3D8
1,  1, 2, 3, 4, 5, 6, 7, 8
**
*ELSET, ELSET=LOADING
1
**------------------------------------------------------------
**  Define fixed boundary conditions
**
*BOUNDARY
1, 1, 3
2, 2, 3
3, 3
4, 1
4, 3
5, 1, 2
6, 2
8, 1
```

```
**-------------------------------------------------------
**  Define the material properties
**
*SOLID SECTION, ELSET=LOADING, MATERIAL=POWDER
*MATERIAL, NAME=POWDER
*USER MATERIAL, CONSTANTS=15
**  Material parameters for aluminium powder
.162E9,5.04,.012E9,8.15,.415,.922,-1.115,100.E6,
**mu1, mu2, lam1,lam2, a,   b,   c,   d,
7.,.52,1., -877.7E6,9.,5.578E4,7.780
**f,RHO0,moduli, g,    h,   j,   k
**
**  Material parameters for iron powder
**6.5385E10,1.4167E11,.3,0.,.9958,-.4191,-.5766,250.E6,
**  G,    K,   nu, , a,   b,   c,   d,
**10.,.4,-1., -3000.E6,7.5,3.371583E6, 6.083256
**f,RHO0,moduli, g,    h,   j,   k
**
**  Moduli indicator
**  1=aluminium (Watson)
** -1=general (Ramakrishnan)
**
*DEPVAR
15
**  The number of state variables (DERVAR) is
**
** 3-D: 15
** AXI: 11
**
**-------------------------------------------------------
**
************************************************************
**  Define analysis type and loading
************************************************************
**
*STEP, INC=1000,UNSYMM=YES
*STATIC
.001, 1., ,.1
**
*EL PRINT, ELSET=LOADING, FREQ=1
S11,S22,S33,SDV13, SDV14, SDV15
*EL PRINT, ELSET=LOADING, FREQ=1
E11,E22,E33
*NODE PRINT, FREQUENCY=0
**
*BOUNDARY,TYPE=DISPLACEMENT
**  Hydrostatic compaction displacements
SIDE1,1,1,-1.E-2
SIDE2,2,2,-1.E-2
SIDE3,3,3,-1.E-2
*RESTART, WRITE, FREQ=1
**
*END STEP
**    End of Analysis
```

APPENDIX H

ABAQUS INPUT FILE: IRON RING COMPACTION

```
*HEADING
INPUT FILE USED FOR RING COMPACTION SIMULATION
**
**
**----------------------------------------------------------
**  Define corner nodes
**
*NODE
1,  0.02,  0.
6, .03625,  0.
61, 0.02,  0.03
66, .03625,  0.03
**
**----------------------------------------------------------
**  Generate the other nodes using node sets and node filling
**
*NGEN, NSET=LEFT
1, 61, 6
*NGEN, NSET=RIGHT
6, 66, 6
*NFILL
LEFT,  RIGHT,  5,  1
**
**----------------------------------------------------------
**  Define master element
**
*ELEMENT, TYPE=CAX4
1,  1, 2, 8, 7
**
**----------------------------------------------------------
**  Generate the other elements using the master element
**
*ELGEN, ELSET=ALL
1,  5, 1, 1,  10, 6, 5
**
**----------------------------------------------------------
**
**  Rigid surface node definition
**
*NODE
999, .05, .015
**
**----------------------------------------------------------
**  Generate the rigid surface
**
*RIGID SURFACE, ELSET=CONTACT1, TYPE=SEGMENTS
```

```
START, .03625, -0.005
LINE, .03625, .035
**
**----------------------------------------------------------
**  Generate contact elements
**
*ELEMENT, TYPE=IRS21A, ELSET=CONTACT1
100, 12, 6, 999
*ELGEN, ELSET=CONTACT1
100, 1, 0, 0, 10, 6, 1
**
**----------------------------------------------------------
**  Define interface
**
*INTERFACE, ELSET=CONTACT1
*FRICTION
0.05
**
**----------------------------------------------------------
**  Second rigid surface node definition
**
*NODE
1000, .03, .05
**
**----------------------------------------------------------
**  Generate the rigid surface
**
*RIGID SURFACE, ELSET=CONTACT2, TYPE=SEGMENTS
START, .0435, 0.03
LINE, .015, .03
**
**----------------------------------------------------------
**  Generate contact elements
**
*ELEMENT, TYPE=IRS21A, ELSET=CONTACT2
110, 61, 62, 1000
111, 62, 63, 1000
112, 63, 64, 1000
113, 64, 65, 1000
114, 65, 66, 1000
**
**----------------------------------------------------------
**  Define second interface
**
*INTERFACE, ELSET=CONTACT2
*FRICTION
0.05
**
**----------------------------------------------------------
**  Rigid surface node definition
**
*NODE
1001, .01, .015
**
**----------------------------------------------------------
**  Generate the rigid surface
**
*RIGID SURFACE, ELSET=CONTACT3, TYPE=SEGMENTS
```

```
START, .02, .035
LINE, .02, -.005
**
**------------------------------------------------------------
**  Generate Contact Elements
**
*ELEMENT, TYPE=IRS21A, ELSET=CONTACT3
120, 1, 7, 1001
*ELGEN, ELSET=CONTACT3
120, 1, 0, 0, 10, 6, 1
**
**------------------------------------------------------------
**  Define interface width
**
*INTERFACE, ELSET=CONTACT3
*FRICTION
0.05
**
**------------------------------------------------------------
**  Rigid surface node definition
**
*NODE
1002, .03, -.01
**
**------------------------------------------------------------
**  Generate the rigid surface
**
*RIGID SURFACE, ELSET=CONTACT4, TYPE=SEGMENTS
START, .015, 0.
LINE, .04125, 0.
**
**------------------------------------------------------------
**  Generate contact elements
**
*ELEMENT, TYPE=IRS21A, ELSET=CONTACT4
150, 2, 1, 1002
*ELGEN, ELSET=CONTACT4
150, 5, 1, 1
**
**------------------------------------------------------------
**  Define interface width
**
*INTERFACE, ELSET=CONTACT4
*FRICTION
0.05
**
**------------------------------------------------------------
**
**  Generate the node set for the lower boundary condition
**
*NSET, NSET=BOUNDARY
1, 2, 3, 4, 5, 6
**
**------------------------------------------------------------
**  Define the boundary conditions
**
*BOUNDARY
**BOUNDARY, 2
```

```
999, 1, 6
1000, 1, 1
1000, 6, 6
1001, 1, 6
1002, 1, 6
**
**------------------------------------------------------------
**
**  Element set for output
**
*ELSET, ELSET=OUTPUT
5, 31, 60
**
**------------------------------------------------------------
**  Define the material properties for new model
**
*SOLID SECTION, ELSET=ALL, MATERIAL=POWDER
*MATERIAL, NAME=POWDER
*USER MATERIAL, CONSTANTS=15
6.5385E10,1.4167E11,.3,0.,.9958,-.4191,-.5766,250.E6,
**nu1, nu2, lam1,lam2, a,   b,   c,   d,
10.,.4,1., -3000.E6,7.5,3.371583E6,6.083256
**f,RHO0,moduli, g,  h,   j,     k
**
**  Moduli indicator
**  1=aluminium (Watson)
** -1=general (Ramakrishnan)
**
*DEPVAR
11
**  The number of state variables (DERVAR) is
**
**  3-D: 15
**  AXI: 11
**
**  Define the material properties for
**  Porous Metal Plasticity Model
**
**SOLID SECTION, ELSET=ALL, MATERIAL=POWDER
**MATERIAL, NAME=POWDER
**ELASTIC
**170E9, .3
**PLASTIC
**2.E7, 0.
**4.E7, .22
**8.5E7, .42
**1.7E8, .55
**3.05E8, .69
**4.65E8, .84
**5.15E8, .94
**POROUS METAL PLASTICITY, RELATIVE DENSITY=.4
**1.5, 1., 2.25
**
**------------------------------------------------------------
**
**************************************************************
**  Define analysis type and loading
**************************************************************
```

```
**
*STEP, INC=20000, NLGEOM,UNSYMM=YES
*STATIC
.001, 1., ,.1
*BOUNDARY, TYPE=DISPLACEMENT
1000, 2, 2, -15.4E-3
**
*EL PRINT, ELSET=OUTPUT, FREQ=0
S22, PRESS, MISES, SDV1, SDV2
**-------------------------------------------------------------
**
*RESTART, WRITE, FREQUENCY=1
*NODE PRINT, FREQUENCY=0
*END STEP
**
**
**
**
**  End of Analysis
```

APPENDIX I

DIMENSIONED DRAWING OF IRON RING

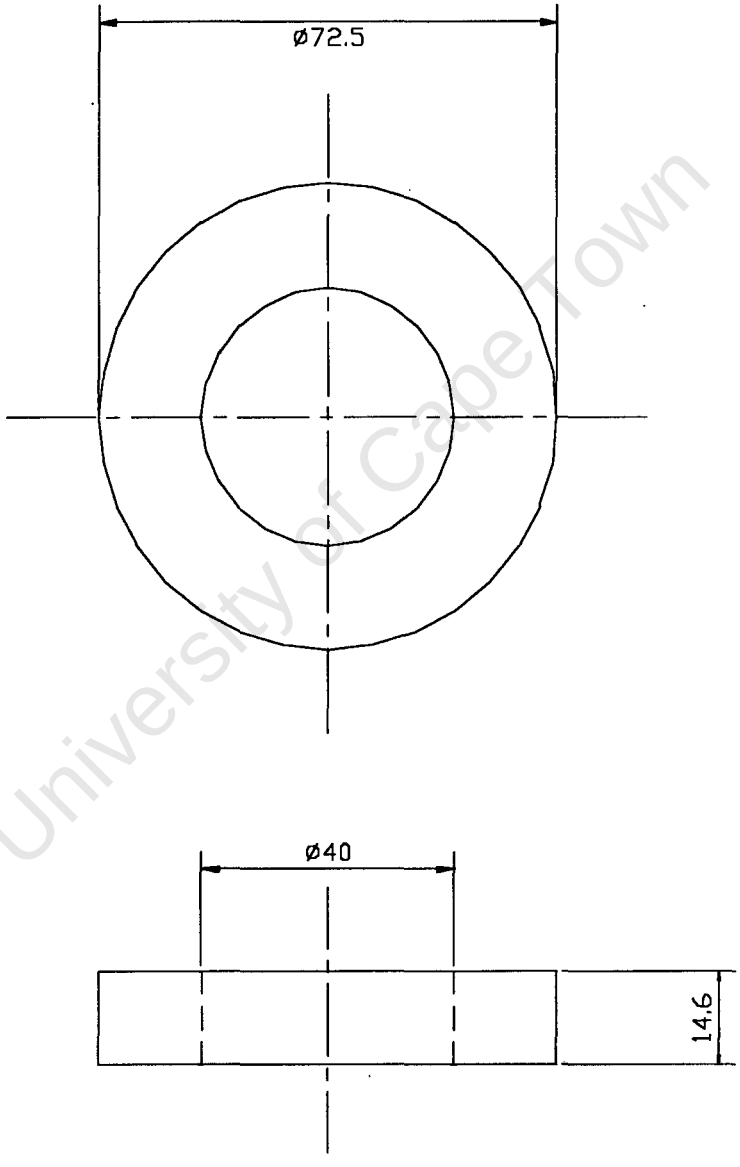

Figure I.1: Final dimensions (mm) of the iron ring.